THE DESIGN AND ANALYSIS OF

INTEGRATED NAVIGATION SYSTEMS

USING REAL INS AND GPS DATA

THESIS
Curtis D. Evans
Captain, USAF

AFIT/GE/ENG/94D-08

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/94D-08

THE DESIGN AND ANALYSIS OF

INTEGRATED NAVIGATION SYSTEMS

USING REAL INS AND GPS DATA

THESIS
Curtis D. Evans
Captain, USAF

AFIT/GE/ENG/94D-08

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or

position of the Department of Defense or the U. S. Government.

Accesion For		
NTIS CRA&I	☒	
DTIC TAB	☐	
Unannounced	☐	
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and/or Special	
A-1		

THE DESIGN AND ANALYSIS OF INTEGRATED NAVIGATION SYSTEMS

USING REAL INS AND GPS DATA

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Curtis D. Evans, B.S. Electrical Engineering

Captain, USAF

December 1994

Preface

I feel very fortunate to have received this thesis topic. This research was of interest to me and made the many hours of work more enjoyable. My sponsor was the Avionics Directorate at Wright-Patterson AFB, OH. I am elated that my next assignment is to my sponsoring organization. Luckily, I will get to use my degree and continue working on similar navigation projects at the Avionics Laboratory.

I would like to extend a special thanks to Lt Col Riggins. His enthusiasm and knowledge of navigation made the research enjoyable. I would like to thank him for always making time in his extremely busy schedule to listen and offer advice. It seemed that every time I reached an insurmountable impasse, he would give me insight into a method of solving the problem.

I also would like to thank my committee members, Dr. Peter Maybeck and Captain Ron Delap, for proof-reading and evaluating my work. In addition, I would like to thank the LaTex guru and my Navigation Laboratory partner, Lt Brian Bohenek, for his support.

Lastly and most importantly, I want to thank my family. My wife, Shonda, has been her usual understanding and patient self over the past several months. She is always there to give me strength and the will to go on to better things. Austin, my son, thank you for letting daddy study when you really wanted to play baseball or Sega. God has truly blessed my life.

<div align="right">Curtis D. Evans</div>

Table of Contents

List of Figures

List of Tables

AFIT/GE/ENG/94D-08

Abstract

An extended Kalman filter (EKF) is used to combine the information obtained from a Global Positioning System (GPS) receiver and an Inertial Navigation System (INS) to provide a navigation solution. This research compares the results of an actual tightly-coupled GPS/INS integrated system with those obtained in computer simulation. Differences between the simulation and actual hardware results, such as navigation solutions and tuning values, are shown in the research. An analysis of various GPS models was studied to determine the best representation for a system truth model . Measurement updates and their effects upon position accuracies was considered and preliminary research was conducted to determine when a feedback implementation was advantageous over a feedforward approach. The results suggest that the computer simulations being used at the Air Force Institute of Technology (AFIT) produce better navigation solutions than are actually possible with real hardware. Alternate system truth models were found which would estimate the errors of an actual system more accurately with less computational loading.

THE DESIGN AND ANALYSIS OF INTEGRATED NAVIGATION SYSTEMS USING REAL INS AND GPS DATA

I. Introduction

The United States Air Force (USAF) currently employs a multitude of navigation systems in its arsenal of weapons. The technological advances of many onboard systems now require more accurate positioning data, which has led the USAF in a continuing search for better navigation solutions. This research focuses on the integration of the Inertial Navigation System (INS) and the Global Positioning System (GPS). The INS has become a standard aircraft navigation tool due to immunity to jamming and relatively accurate positioning. The Global Positioning System is rapidly growing in popularity in navigation systems, and the USAF has committed itself to installing GPS systems in all of its aircraft. The main objective of integrating these two systems is to combine position estimates from each system and produce an even more accurate navigational solution.

This thesis is a natural progression in the study of integrated navigation systems at the Air Force Institute of Technology (AFIT) (10, 11, 18, 20, 22, 27). The goal of this research is to verify the models and methods used by AFIT in past studies by comparing simulation results to the position estimates of actual tightly-coupled integrated navigation systems. Section 1.2.5.1 discusses the tightly-coupled approach in GPS/INS integrated navigation systems. The navigation systems integrated in this thesis are NAVSTAR XR-4PC and XR-5PC Global Positioning System Receivers and a Litton LN-94 Inertial Navigation System. The research has been sponsored by the Avionics Laboratory, Wright Patterson AFB, OH.

In the not too distant future, the AFIT Navigation Section hopes to develop a mobile GPS/INS laboratory. Although this thesis pertains only to stationary platforms in a post-processing

environment, the results will provide the groundwork for the development of a real-time GPS/INS integrated system on a moving vehicle.

1.1 Background

AFIT navigation research has covered a myriad of subjects concerning the integration of navigation systems; the vast majority of these theses were done in simulation (10, 18, 20, 22, 27). The research began with the generation of computer models for the INS, GPS, and Range/Range-Rate System (RRS) (22). These models were primarily used in theses supporting the navigation reference systems utilized by the Central Inertial Guidance Test Facility (CIGTF) at Holloman AFB, NM. AFIT selected the extended Kalman filter (EKF) method in most research efforts to combine the navigation information. These investigations delved into the performance of various navigation systems, as well as several different methods of failure detection, isolation, and recovery (10, 18, 27). The computer simulation models and methods from past theses are used as a basis for this research. The Multimode Simulation for Optimal Filter Evaluation (MSOFE) models used in past AFIT research has incorporated GPS satellite pseudorange, Range/Range-Rate transponder, barometric altimeter, and Doppler radar measurements (10, 18, 3, 20, 27). This thesis will extract the INS/GPS portions of these system truth and filter models and employ only GPS pseudoranges for measurements. Pseudoranges, pseudorange-rates, and carrier-phase measurements are available from the laboratory equipment. In order to limit the scope of the research, only pseudorange measurements were used in this initial attempt of a GPS/INS hardware integration at AFIT.

Nearly all AFIT navigation research has been conducted in simulation only. One notable exception was an attempt in 1990 to accomplish a hardware integration of a Collins 3A GPS receiver and an LN-94 INS (11). The information necessary to integrate the two systems in a tightly-coupled manner was not readily available from the communications port of the Collins 3A receiver. To this researcher's advantage, the XR-4PC and XR-5PC have the pseudorange and

satellite ephemeris data readily accessible. The benefits of raw data access will be discussed in the Literature Review.

1.2   Literature Review

1.2.1   Navigation.   This section provides background information on the individual subsystems and the reasoning supporting their selection for integration in the EKF. Navigation is defined as the determination of a physical body's position and velocity relative to some reference frame (1). The INS, GPS, and EKF are the primary components in this integrated navigation research. The outputs of the INS and GPS are combined in an extended Kalman filter to provide the navigation solution. In order to understand the integration design process fully, the operation of each of these subsystems will be discussed in the ensuing sections.

1.2.2   Inertial Navigation System.   The inertial navigation system (INS) utilizes gyroscopes and accelerometers mounted aboard the vehicle to execute the navigation function. An appropriately initialized inertial navigation system is capable of continuous determination of vehicle position and velocity without the use of external radiation or optical information. The INS contains gyroscopes which compute or maintain the reference coordinate frame, and accelerometers which measure specific force. Specific force is the measurement of acceleration minus the gravity vector. In order for the specific force to be computed accurately, the gravity vector must be modeled properly. By separating the effects of gravity, the vehicle accelerations can be precisely determined. The inertial navigation unit integrates the acceleration twice to obtain position measurements. All inertial navigation systems perform the following functions: instrument or compute a reference frame, measure specific force, have knowledge of the gravitational field, and integrate the specific force data in time to obtain velocity and position information (1).

The inertial navigation system to be used in this thesis research is a Litton LN-94 ring laser gyro strapdown system. The maximum drift for the LN-94 is approximately one nautical

mile per hour circular error probability (CEP). The LN-94 transmits/receives information over the 1553 data bus (5). The LN-94 utilizes six different reference frames: earth-centered earth-fixed (ECEF), navigation, true, computer, platform, and body (8). Since the gyros and accelerometers are physically mounted to the frame of the aircraft, the measurements are taken in the body frame. The INS integrates the specific force measurements and converts the results into the navigation frame. The truth model used in simulation is a derivation of the Litton 93-state model given in the LN-93 error budget (8). The LN-93 and LN-94 are internally the same guidance system, the only difference being the packaging of the individual units. Therefore, the truth model applies to both inertial navigation units.

1.2.3   Global Positioning System.   The Global Positioning System provides three-dimensional position and velocity data to users anywhere in the world. The GPS system consists of three segments: space, control, and user. The space segment is made up of 24 satellites in six orbital planes which receive information from the control segment and transmit satellite orbital information to the user segment. This particular constellation of satellites is configured so that a user has at least four satellites visible anywhere in the world at any time with only a few brief outages at remote locations. The control segment monitors satellites and performs updates when necessary. These updates include clock corrections, new ephemerides, and command telemetry (2). The user equipment must receive signals from at least four different satellites in order to provide three-dimensional position and velocity data to the user. The computer simulation uses four satellites placed in orbit to produce a desired geometry. When collecting data for the hardware integration, all satellites in view will be used for measurements. All satellites in view will be used in order to provide the best possible Geometric Dilution Of Precision, GDOP.

The pseudoranges and ECEF satellite positions are the primary GPS sensor measurements used in the extended Kalman filtering algorithm. Pseudorange is the time shift required to line up a replica of the code generated in the receiver with the received code from the satellite, multiplied

by the speed of light. Ideally, the time shift is the difference between the time of signal reception (measured in the receiver time frame) and the time of emission (measured in the satellite time frame) (2). These two time frames are different, which introduces a bias into the measurement. These biased time delay measurements are thus referred to as pseudoranges.

The integrated navigation system concentrates mainly on the user segment. The GPS user equipment is comprised of four principal components: antenna, receiver, computer, and input/output devices (2). The user equipment used in this thesis will be the NAVSTAR XR-5PC and XR-4PC GPS receivers. These receivers are capable of providing raw data, such as pseudorange, delta-pseudorange, and ephemeris data, as well as the GPS receiver's Kalman filter-computed user position in the navigation frame. Both receivers are capable of receiving the unclassified GPS code only; therefore, selective availability is a factor in determining range accuracies. Selective availability is the intentional degrading of the GPS signal by the control segment. Only users authorized by the military are capable of receiving the classified signal. The XR-5PC is a six-channel receiver providing information on up to eight satellites in view (26). The XR-4PC is also capable of tracking eight satellites; however, this receiver only contains two channels (25).

1.2.4 Extended Kalman Filter Overview. A Kalman filter is an optimal recursive data processing algorithm (14). The conventional Kalman filter is based upon linear models. Because the navigation equations are nonlinear and Kalman filters require linear models, a variation from the traditional Kalman filtering technique is necessary. The type of estimator used in this navigation integration scheme is an extended Kalman filter. The extended Kalman filter relinearizes about each estimate once it has been computed. By using the extended Kalman filter, validity of the assumption that deviations from the nominal trajectory are small enough to allow linear perturbation techniques to be employed is improved (15). Rather than relying solely on either the INS or the GPS navigation solution, the Kalman filter uses the statistical characteristics of the errors in both systems to determine the optimal combination of information.

1.2.5 GPS/INS Integration.   The GPS/INS integration exploits the individual strengths of each system. The advantage of GPS is long-term accuracy, which can be used to upgrade the performance of an INS. The INS has the weakness of slowly drifting over a long period of time. In other words, the filter uses the good low frequency information from the GPS data to damp out slowly growing errors inherent in the INS. On the other hand, the INS has exceptional qualities in the short term and can be used to enhance the performance of GPS, especially in a highly dynamic flight scenario or a jamming environment. The GPS/INS signals may be optimally combined to provide even greater navigational accuracy, or in the event of a failure, each system may be used independently. These integrations provide a system that exploits the complementary characteristics of both GPS and INS.

1.2.5.1 Integration Approaches.   There exist several methods of integrating GPS and INS information via an extended Kalman filter. The Kalman filter may use a direct (total state space) or indirect (error state space) implementation, and feedforward or feedback mechanizations. The direct approach employs total states such as vehicle position and velocity in the filter. The direct implementation places the filter directly into the INS control loop; therefore, the filter would have to keep up with vehicle angular motion and be required to suppress noisy and sometimes erroneous data. Most INS sampling rates range from 50-100 Hz; this sampling rate requirement makes this type of integration unfeasible for in-flight applications. The Kalman filter requires too much computation time to keep pace with the sampling rate that would be required in following the movement of the vehicle. In addition, some of the vehicle dynamics are nonlinear, so a conventional linear Kalman filter could not be used. Yet another disadvantage is that if the filter should happen to fail, the INS cannot operate (14).

The indirect approach estimates the errors in the navigation and attitude information using the difference between the INS and GPS data. Since the filter is out of the INS loop, the matter of reliability or through-put is no longer an issue. Also, the sampling rate for an indirect

implementation is much slower since the filter follows the lower frequency *errors* of the INS rather than the aircraft dynamics (14). Sample periods on the order of a number of seconds are not atypical. This slower sampling rate provides the extended Kalman filter with much more time to complete its computations.

The feedforward approach compares the GPS and INS data and uses the result to estimate the errors in the inertial system. By subtracting the estimated errors from the inertial data, the computer maintains the optimal estimates of position, velocity and attitude. In this configuration, the INS operates independently. In other words, the INS errors will continue to slowly grow. As previously mentioned, the Kalman filter is based on the adequacy of its linearized model, so it is necessary for the errors in the inertial system to remain of small magnitude (14). The integration designs developed at AFIT over the past several years have been of this indirect feedforward type. The feedforward implementation has worked well in AFIT simulations because linear truth models are used. Since the truth and filter models are both linear, large INS errors can be accurately estimated by the filter. However, in the real world, the actual system error dynamics are nonlinear, so linear models are unable to estimate large INS errors properly. The extended Kalman filter estimates are only as good as the models being employed. This research will explore the adequacy of the computer simulation's linear truth models in a feedforward configuration versus the actual GPS/INS outputs. Figure 1.1 illustrates a feedforward implementation.

The Kalman filter indirect feedback mechanization generates the estimates of the errors in the inertial system, but these are fed back to the INS as corrections. This method does not allow the inertial errors to grow unchecked, and the adequacy of the linearized model is enhanced. Over the long term, the indirect feedback implementation is more accurate than the feedforward approach. A disadvantage of the feedback method is that the INS is totally dependent on the EKF estimates. This reliance on the EKF permits the INS to be reset to erroneous positions due to glitches in the EKF output. The use of long sample periods in a feedback implementation allows
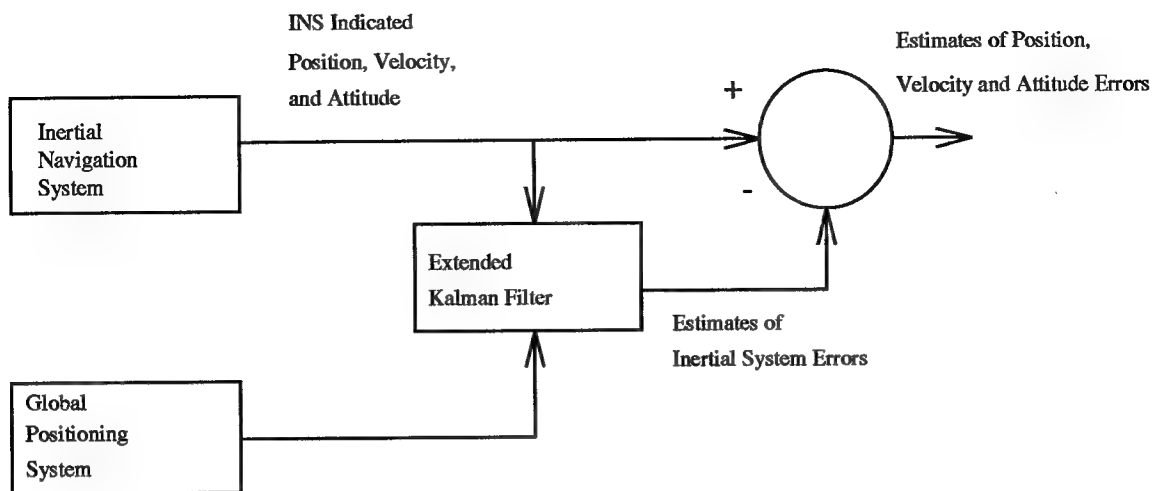
Figure 1.1   EKF Feedforward Implementation for a GPS/INS Integration

sufficient time to test the EKF's navigation solution before applying it. The system employing the

feedforward method could provide three separate solutions: GPS, INS, and GPS/INS navigation

solutions. The feedback implementation can only provide a GPS and GPS/INS solution. One of

the goals of this research is to discern when a feedback implementation becomes more advantageous

than the feedforward approach. Figure 1.2 shows the structure of a feedback implementation.

This thesis research will use a tightly-coupled scheme for the GPS/INS integration. A

tightly-coupled implementation was chosen for this research because it provides the more optimal

navigation solution and has been used by AFIT in past theses (10, 18, 20, 22, 27). *Tightly-coupled*

systems, use the most basic of information from each sensor instead of an output from the sensor's

local filter. For instance, satellite ephemeris data is collected to compute ECEF positions, and

pseudoranges are taken directly from the satellites. Other integrations, called *loosely-coupled*, use

the position outputs of a Kalman filter solely dedicated to the GPS unit, rather than raw data

sent to the GPS/INS integrated system's extended Kalman filter. The instability problem caused

by filters driving filters enters the picture with loosely coupled systems, because the errors in

various sensors are cross-correlated. The cross-correlation and temporal correlation of data are

not adequately modeled or compensated for in the GPS/INS integrated filter's design. Techniques
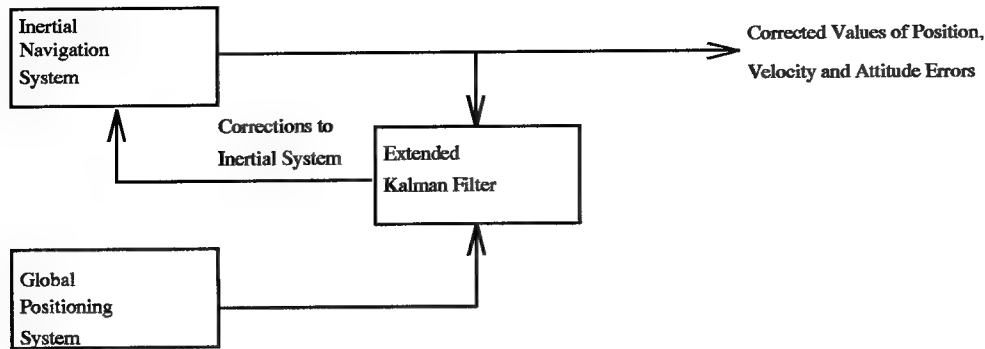
Figure 1.2   EKF Feedback Implementation for a GPS/INS Integration

have been developed to use this preprocessed data, most of which reduce the temporal correlation of data by lengthening the measurement sampling time (11, 24). The technique of using longer sampling times addresses the temporal correlation issue, but does not confront the cross-correlation issue. This thesis will only concentrate on tightly-coupled systems, and will not pursue the issues of loosely-coupled systems.

The integration of components is the chief technical obstacle to GPS/INS optimal integration, or tight-coupling. The benefits to be gained from optimal integration or tight-coupling of GPS and INS components are high performance in terms of navigation accuracy, rejection of radio interference and dynamics tracking ability, and low cost in terms of total electronics package size, weight, power consumption and production cost. As previously stated, the GPS/INS combination can function more efficiently in severe environments than either system alone. Inertial aiding to the GPS receiver allows it to use narrower tracking loops for rejecting jamming and for rapid re-acquisition after the interruption of signals. Conversely, the inertial navigation system will only navigate accurately for a short period of time without in-flight resets from an integration filter using GPS measurements (12).

Although the advantages are very attractive, tightly-coupled systems do have potential problem areas. Timing difficulties are the primary problem areas. The GPS and INS outputs must be synchronized with a common timing signal in order to provide appropriate input data to the

EKF. If the timing signal is lost, then the integrated navigation solution is lost (24). The timing issue is somewhat diminished in this research due to the fact that the platform remains stationary; therefore, matching GPS and INS outputs with vehicular movement is irrelevant.

1.3 Problem Definition

The research conducted under this thesis will employ both full and reduced order models developed in past AFIT theses and Avionics Directorate of Wright Laboratory research. These models will be used in an extended Kalman filter to provide a navigation solution. In order to determine the accuracy of these models, the navigation solutions of different computer simulations will be compared to the GPS/INS integrated hardware results. In a perfect world, the simulation outputs would mirror the real world. Realistically, we know this will not be the case; however, a subject of particular interest is to determine exactly how close the computer simulation is to the actual results.

In addition to computer simulation versus real world, several different filters will be used in the EKF of the hardware integration and their performances compared against each other. The purpose of this comparison is to determine the accuracy of the models in obtaining a navigation solution. The filters studied will be the 69-state, 61-state, 49-state, 41-state, and 13-state; Chapter III describes these filters in detail.

This research will use an extended Kalman filter with an indirect implementation. The system is tightly-coupled and feedforward. The results will be analyzed to determine the consequences of not resetting the INS, as in a feedback configuration.

1.4 Scope

This thesis will concentrate on a stationary platform. The scope is defined by the limitations imposed by the AFIT Navigation Laboratory facilities. The current capabilities of the

laboratory do not include a method to operate the LN-94 INS on a moving platform. The models developed by AFIT were used in conjunction with a computer-simulated flight profile produced on the software package PROFGEN (19). The stationary platform restriction imposed by the hardware somewhat limits the results of this research, but does establish a starting point for future research. Due to the unobservability of some states while in a stationary position, the results of this research may not fully apply to moving platforms, especially to filter tuning results. This GPS/INS integration will be done in a post-processing mode. Eventually, AFIT hopes to have a real-time tightly-integrated system. Since this research is a pioneering effort and has not previously been accomplished at AFIT, post-processing was chosen as a first step. The results of this thesis should provide a foundation for future GPS/INS integrations.

1.5 Assumptions

Assumptions are an integral part of any study so that the results can be accurately evaluated. This section outlines the assumptions that have been made in this thesis.

1. The computer simulations were developed and operated using routines written in the commercially available software, MATLAB (13). The MATLAB routines were directly patterned after the software Multimode Simulation for Optimal Filter Evaluation (MSOFE) (3). MSOFE is an established USAF software package to develop and test Kalman filter algorithms. The MATLAB routines were verified by running identical simulations on both MATLAB and MSOFE and comparing the results (7).

2. The MATLAB simulations were conducted using 10-run Monte Carlo analyses. While a larger run size for the Monte Carlo analysis is preferable, this number of runs was selected due to computer and software limitations.

3. The INS platform is assumed to be stabilized with a barometric altimeter. An INS is unstable without an outside measurement source in the vertical channel. While a barometric altimeter

is not the only way to stabilize a platform, it is a commonly used method for the LN-94. The use of the barometric platform is included in the modelling of the system. The simulated output of a barometric altimeter was sent to the LN-94 over the 1553 data bus to stabilize the vertical channel during data collection (5). Unlike past AFIT theses, the barometric altimeter will not be used as an explicit measurement for the EKF, since the actual barometric altimeter is not available in the laboratory. An objective of the research is to use only sensor measurements in the simulation that are available in the laboratory. The sensor measurements available in the Navigation Laboratory are GPS pseudoranges.

4. The stationary platform analysis developed in this research will be extendable to a moving platform in future AFIT research.

1.6   Plan of Attack

This research is broken down into three basic components: GPS/INS integrated system simulation, hardware integration of the XR-5PC or XR-4PC and the LN-94, and the comparison of results between simulation and the actual integrated system .

1.6.1   GPS/INS Integrated System Simulation.   The first task is to integrate the GPS and INS in computer simulation. The simulation is to be conducted using a collection of MATLAB m-files patterned after the USAF's Multimode Simulation for Optimal Filter Evaluation (MSOFE). The m-file allows the user to execute sequences of commands that are stored in files with names that have an extension of .m (13). A description of these m-files, called MATSOFE, is contained in Appendix J. Unlike past AFIT simulations conducted for the Central Inertial Guidance Test Facility (CIGTF) at Holloman AFB, the only systems integrated will be the GPS and INS. Past AFIT theses have used the barometric altimeter, Doppler radar, and Range/Range-Rate systems for additional measurements. The previous models used in MSOFE have strangely required the use of the perfect Doppler radar measurements for velocity aiding (10, 18, 20, 27). These perfect velocity

measurements appear to be a coding problem in MSOFE and in theory are not necessary. This dilemma will be confronted and solved so that only GPS and INS measurements will be required. In the hardware integration, a simulated barometric altimeter measurement will be used to stabilize the INS vertical channel. In reality, the INS uses the barometric altimeter solely for vertical channel stability and the altimeter measurement for the extended Kalman filter should not be necessary to accomplish accurate position estimates. Once the simulation is running satisfactorily, the reduced order filter used in the EKF will be tuned and the results plotted for later comparison.

1.6.2 GPS/INS Hardware Integration. The second step in this research will be to integrate the XR-4PC or XR-5PC and the LN-94. The basic MATLAB m-files used in the computer simulation will be altered to accept real data. The data will be collected from the LN-94 and GPS receivers and combined in the EKF. The data files will be used as input for the EKF in a post-processing methodology. The filter will be retuned for the real data and the results noted for comparison. Data from both the XR-4PC and XR-5PC will be collected to determine the differences in two-channel and six-channel receivers.

1.6.3 Comparison of Results. The final step is to analyze the results of the data collected from the hardware integration and the computer simulation. Specific areas of interest are the validation of the 39-state LN-94 truth model and various GPS truth models using real data; the optimal navigation accuracy attainable using the various GPS filter models in a hardware integration; and investigation of actual integration versus simulation results. In both simulation and hardware integration, the full order truth model and reduced order models will be used in the filter. In addition, the navigation solutions from the inertial navigation system, extended Kalman filter, and the GPS batch least squares computations using real data will be compared. The research will delve into issues such as measurement update rates and the validity of the feedforward implementation.

1.7 Overview of Thesis

Chapter II presents the theory used in this research. Extended Kalman filter theory is introduced first. In addition, the topics of filter order reduction and tuning will be discussed. The theory used to develop the algorithms for calculating GPS satellite ECEF positions is presented. The chapter concludes with a presentation of batch least squares processing used in determining a position solution with GPS data only.

Chapter III describes in detail the navigation system parameters and the INS and GPS system truth models used. An overall system description is defined for both the computer simulation and the hardware integration. The final section of the chapter presents the EKF measurement equations and filter models.

Chapter IV presents the results of this study. A comparison of the simulation system truth and reduced order filter models to the actual hardware integration results is shown. In addition, the performance of the individual subsystems is compared with the integrated navigation system results. Finally, results are presented which provide a basis as to when a feedback implementation becomes the more viable method in comparison to the feedforward implementation.

Chapter V summarizes the research with conclusions and recommendations for further research. The recommendations are specifically aimed at the goal of establishing a real-time, mobile GPS/INS integrated navigation system at AFIT.

II. Theory

2.1   Overview

The purpose of this chapter is to provide an overview of the theory necessary to accomplish this research. The first section presents Kalman filtering theory, followed by a discussion on model reduction and concluding with the subject of filter tuning. Complete derivations and discussions of the noted topics can be found in the references (14, 15). The theory then moves into the area of determining GPS satellite ECEF positions. The chapter ends with a presentation of batch least squares theory in computing GPS-only derived navigation solutions.

2.2   Extended Kalman Filter Equations

A Kalman filter is an optimal recursive data processing algorithm which is based upon linear models. The GPS receiver and INS error state models consist of a set of linear state-space differential equations and nonlinear measurement equations. These nonlinearities prevent the use of a linear Kalman filter. Because of this constraint an extended Kalman filter (EKF) is to be implemented in this project. The basic idea of the EKF is to *relinearize* about each state estimate once it has been computed (15). This relinearizing about newly declared nominals at each sample time enhances the adequacy of the linearization process. The subsequent derivation and many of the following equations are taken from Maybeck (15).

The state models are a set of nonlinear continuous-time differential equations of the form:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.1}$$

where $\mathbf{f}[\mathbf{x}(t), t]$ is the state dynamics vector which in general is a nonlinear function of the state vector $\mathbf{x}(t)$ and of time t. $\mathbf{G}(t)$ is a noise distribution matrix which for this research is an identity ($\mathbf{I}$) matrix without loss of generality. The vector represented by $\mathbf{w}(t)$ is a white Gaussian noise

vector process with the following statistics:

$$\mathbf{m}_w = E\{\mathbf{w}(t)\} = \mathbf{0} \tag{2.2}$$

and noise strength:

$$E\{\mathbf{w}(t)\mathbf{w}^T(t+\tau)\} = \mathbf{Q}(t)\delta(\tau) \tag{2.3}$$

The measurements of the system are assumed to be discrete-time measurement updates of the form:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \tag{2.4}$$

where $\mathbf{z}(t_i)$ is the measurement available at time t_i, $\mathbf{h}$ is a known vector which is a function of the states and time. The vector $\mathbf{h}$ can be either linear or nonlinear. For this effort, $\mathbf{h}$ is a nonlinear function of the state vector and time due to the nonlinear nature of the GPS measurements. The vector $\mathbf{v}(t_i)$ represents a white Gaussian noise process with the following statistics:

$$\mathbf{m}_v = E\{\mathbf{v}(t_i)\} = \mathbf{0} \tag{2.5}$$

with noise covariance:

$$E\left\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\right\} = \begin{cases} \mathbf{R}(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases} \tag{2.6}$$

For the EKF to produce an estimate of the error state vector $\mathbf{x}(t)$, the system must first be linearized. To form the linearized perturbation equations, the linearization method described in (14) will be used. The following derivation is the linearization of Equations (2.1) and (2.4) using this method.

First the nominal state trajectory, $\mathbf{x}_n(t)$, satisfies the differential equation:

$$\dot{\mathbf{x}}_n(t) = \mathbf{f}[\mathbf{x}_n(t), t] \tag{2.7}$$

which starts from the initial condition $\mathbf{x}_n(t) = \mathbf{x}_{no}$ where $\mathbf{f}(\cdot, \cdot)$ is the same as defined in equation (2.1). The nominal noise-free measurement equation taken with respect to this nominal trajectory becomes:

$$z_n(t_i) = \mathbf{h}[\mathbf{x}_n(t_i), t_i] \tag{2.8}$$

where $\mathbf{h}[\cdot, \cdot]$ is as given in (2.4).

To perturb the actual state from this assumed nominal state trajectory, subtract Equation (2.7) from (2.1):

$$[\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_n(t)] = \mathbf{f}[\mathbf{x}_n(t), t] - \mathbf{f}[\mathbf{x}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.9}$$

Now taking a Taylor series expansion about $\mathbf{x}_n(t)$, $\mathbf{f}[\mathbf{x}, t]$ becomes:

$$\mathbf{f}[\mathbf{x}(t), t] = \mathbf{f}[\mathbf{x}(t), t]|_{\mathbf{x}=\mathbf{x}_n(t)} + \frac{\partial}{\partial x}\mathbf{f}[\mathbf{x}, t]|_{\mathbf{x}=\mathbf{x}_n(t)}\delta\mathbf{x}(t) + h.o.t. \tag{2.10}$$

where $\delta\mathbf{x}$ is the perturbation state given by $[\mathbf{x}(t) - \mathbf{x}_n(t)]$ and $h.o.t.$ is higher order terms which represent terms of $\delta\mathbf{x}$ in powers greater than one. Invoking a first order approximation and substituting Equation (2.10) into Equation (2.9):

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}[t; \mathbf{x}_n(t)]\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.11}$$

where $\delta\dot{\mathbf{x}}(t)$ is the perturbation state defined by $[\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_n(t)]$ and the matrix $\mathbf{F}[t; \mathbf{x}_n(t)]$ is defined by:

$$\mathbf{F}[t; \mathbf{x}_n(t)] = \left.\frac{\partial \mathbf{f}[\mathbf{x}(t), t]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n(t)} \tag{2.12}$$

2-3

Using the same procedure on Equation (2.4), the perturbed discrete-time measurement equation is expressed as:

$$\delta z(t_i) = H[t_i; x_n(t_i)]\delta x(t_i) + v(t_i) \tag{2.13}$$

where the matrix $H[t_i; x_n(t_i)]$ is defined by:

$$H[t_i; x_n(t_i)] = \left. \frac{\partial h[x, t_i]}{\partial x} \right|_{x=x_n(t_i)} \tag{2.14}$$

The nonlinear dynamics and measurement update equations have been linearized to form "perturbation" or "error" state equations. This linearization process allows for the application of a linearized Kalman filter for the system described by Equations (2.11) and (2.13). The filter implemented will output the optimal estimate of the state error vector δx represented by $\hat{\delta x}$. The estimate of the total state of the system, $\hat{x}(t)$, can be computed using:

$$\hat{x}(t) = x_n(t) + \delta\hat{x}(t) \tag{2.15}$$

The preceding derivation is adequate so long as the "true" and nominal trajectories do not differ significantly; if they do, large unacceptable errors will result. This requirement is clearly unreasonable for most navigation scenarios. To avoid this possible problem, an EKF is to be used in this application. The EKF relinearizes about newly declared nominal trajectory segments and the measurement vector about each new state estimate. This redeclaration of the states about the new nominal trajectory ensures that the deviations from the nominal trajectory will remain small. This validates the assumption made earlier and allows for linear perturbation techniques to be employed with adequate results.

The state estimate and covariance are propagated from time t_i to the next sample time t_{i+1} through the integration of the following equations:

$$\dot{\hat{\mathbf{x}}}(t/t_i) = \mathbf{f}[\hat{\mathbf{x}}(t/t_i), t] \qquad (2.16)$$

$$\dot{\mathbf{P}}(t/t_i) = \mathbf{F}[t; \hat{\mathbf{x}}(t/t_i)]\mathbf{P}(t/t_i) + \mathbf{P}(t/t_i)\mathbf{F}^T[t; \hat{\mathbf{x}}(t/t_i)] + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \qquad (2.17)$$

where the notation (t/t_i) stands for "at time, t, based on measurements up through time t_i", and where:

$$\mathbf{F}[t; \mathbf{x}(t/t_i)] = \left.\frac{\partial \mathbf{f}[\mathbf{x}, t]}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}} = \hat{\mathbf{x}}(t/t_i)} \qquad (2.18)$$

using the results of the previous measurement update cycle as initial conditions:

$$\hat{\mathbf{x}}(t_i/t_i) = \hat{\mathbf{x}}(t_i^+) \qquad (2.19)$$

$$\mathbf{P}(t_i/t_i) = \mathbf{P}(t_i^+) \qquad (2.20)$$

and the superscript, +, indicates the value at a time after the incorporation of a measurement.

With the incorporation of discrete-time measurements, z_i, the EKF is accomplished through the following equations:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T[t_i; \hat{\mathbf{x}}(t_i^-)]\{\mathbf{H}[t_i; \hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-)\mathbf{H}^T[t_i; \hat{\mathbf{x}}(t_i^-)] + \mathbf{R}(t_i)\}^{-1} \qquad (2.21)$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\{z_i - \mathbf{h}[\hat{\mathbf{x}}(t_i^-), t_i]\} \qquad (2.22)$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}[t_i; \hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-) \qquad (2.23)$$

where $\mathbf{H}[t_i; \hat{\mathbf{x}}(t_i^-)]$ is defined in Equation (2.14) and the superscript, -, indicates a value at a time before incorporation of a measurement.

2.3   Kalman Filter Order Reduction

The goal of filter order reduction is to decrease the number of states in the filter model with respect to the truth model states without significantly degrading the accuracy of the remaining state estimates. Filter order reduction is an important step in the actual on-line implementation of Kalman filter designs. The use of a full-order filter is computationally burdensome for most aircraft computers due to the large number of truth model states which must be evaluated. Aircraft computer systems have several functions which must run simultaneously; in order to be able to perform all functions adequately on-line and in real-time, filter order reduction must be accomplished. Thus, the reduction in states sacrifices to some extent the performance of the filter. The goal is to eliminate only the states which have a minimal impact on the navigation solution. The restraints on computer time justifies the need for filter order reduction. Although filter order reduction reduces the computational burden, the reduced order filter will provide a suboptimal solution. To compensate for this sub-optimality problem, the process of filter tuning must be accomplished. Filter tuning will be discussed next in Section 2.4.

2.4   Kalman Filter Tuning

The basic objective of filter tuning is to achieve the best possible estimation performance from a particular filter, i.e., totally specified except for $\mathbf{P}_0$, $\hat{\mathbf{x}}_o$, and the time histories of $\mathbf{Q}$ and $\mathbf{R}$. These statistics account for actual disturbances and noises in the system, as well as uncertainties of the filter model due to order reduction and nonlinearities. The less accurate the filter model, the greater the noise strengths or uncertainty in the model must be. In tuning the filter, $\mathbf{P}_0$ determines the initial transient performance of the filter, whereas the $\mathbf{Q}$ and $\mathbf{R}$ histories dictate the longer term steady state performance (14).

In order for the filter to track the given trajectory accurately in the extended Kalman filtering problem, the process and measurement noise strengths $\mathbf{Q}$ and $\mathbf{R}$ must be appropriately tuned. By

increasing the filter process noise strength (**Q**), the uncertainty of the filter model is increased. The added process noise causes the filter to rely more heavily on the information provided by the incoming measurements. An increase in **Q** causes the $\mathbf{P}(t)$ elements to grow more rapidly between measurement times. Conversely, an increase in the observation matrix noise covariance (**R**) results in less confidence in the sensor measurements. As would be expected, an increase in measurement noise causes the filter to place more confidence in the dynamics model. If the eigenvalues of **Q** are large compared to the eigenvalues of **R**, steady state is quickly reached because the uncertainty involved in the state propagation is large compared to the accuracy of the measurements, so the new state estimate is heavily dependent upon the new measurement and not closely related to prior estimates (14).

The MSOFE User's Manual gives a list of criteria to meet when evaluating filter behavior (3). These criteria were used as a guideline in tuning the filters in simulation. The filter tuning criteria emphasized most will be that the filter estimation error should be zero-mean and the mean ± one sigma of estimation error should fall within the envelope formed by zero ± one filter-predicted sigma. Of course, in the case of the hardware integration, the tuning values which provide the best position and velocity solutions will be used.

2.5 Satellite Position Determination

In order to accomplish the GPS/INS tightly-coupled hardware integration, the ECEF positions of the satellites must be computed. These ECEF positions are used to calculate an INS-predicted pseudorange to each satellite. This section will describe the theory used in developing an algorithm to compute the ECEF positions using GPS satellite ephemeris data. The complete development of the theory can be found in a paper written by Van Dierendonck (28).

A Keplerian representation is employed in the GPS system to represent the nominal satellite orbit with additional parameters describing the perturbations. The perturbations about the Keplerian orbit are required to obtain the required accuracy. Kepler's equation is given by:

$$E(t) = M(t) + e \sin E(t) \qquad (2.24)$$

where:

$$E(t) \quad = \quad \text{Eccentric anomaly}$$

$$M(t) \quad = \quad \text{Mean anomaly}$$

$$e \quad = \quad \text{Eccentricity}$$

The equations for the anomaly from the nominal orbit is

$$\sin v(t) = \frac{\sqrt{1 - e^2} \sin E(t)}{1 - e \cos E(t)} \qquad (2.25)$$

$$\cos v(t) = \frac{\cos E(t) - e}{1 - e \cos E(t)} \qquad (2.26)$$

where $v(t)$ is the true anomaly. It is impractical to solve for the nonlinear term, E(t), in any way except approximately because, for e $\leq$ 0.663, the exact solution is

$$E(t) = M(t) + 2 \sum_{k=1}^{\infty} \frac{1}{k!} J_k(ke) \sin[k M(t)] \qquad (2.27)$$

where the J_k terms are Bessel functions of the first kind of order k. In this thesis, the solution was found by using successive substitutions to solve Kepler's equation. Five iterations were found to be sufficient in this circumstance (6).

Table 2.1   Ephemeris Representation Parameters

M_0	Mean anomaly at reference time
Δn	Mean motion difference from computed value
e	Eccentricity
$\sqrt{A}$	Square root of the semi-major axis
Ω_0	Right Ascension at reference time
i_0	Inclination angle at reference time
ω	Argument of perigee
$\dot{\Omega}$	Rate of right ascension
C_{uc}	Amplitude of the cosine harmonic correction term to the argument of latitude.
C_{us}	Amplitude of the sine harmonic correction term to the argument of latitude.
C_{rc}	Amplitude of the cosine harmonic correction term to the orbit radius.
C_{rs}	Amplitude of the sine harmonic correction term to the orbit radius.
C_{ic}	Amplitude of the cosine harmonic correction term to the angle of inclination.
C_{is}	Amplitude of the sine harmonic correction term to the angle of inclination.
t_0	Ephemeris reference time.
AODW	Age of Data Word

The ephemeris information is transmitted to the GPS receiver which describes the satellite orbit for a one hour interval of time. This information also describes the ephemeris for an additional 30 minutes to allow the user to receive the parameters for the new interval of time. The definitions of the parameters are given in Table 2.1. An age of data word (AODW) is provided to the user to give a confidence level in the ephemeris representation parameters. AODW represents the time difference between the reference time (t_0) and the time of the last measurment update (t_l) used to estimate the representation parameters.

$$AODW = t_0 - t_l \tag{2.28}$$

The ECEF positions of the GPS satellites are found by first solving for the mean motion, n_0, using the semi-major axis of the orbit, A, and $\mu = 3.986005x10^{14} meters/second$:

$$n_0 = \sqrt{\frac{\mu}{A^3}} \tag{2.29}$$

The corrected mean motion is then determined by

2-9

$$n = n_0 + \Delta n \qquad (2.30)$$

The time from epoch is computed from the difference in actual time and the ephemeris reference time:

$$t_k = t - t_0 \qquad (2.31)$$

The mean anomaly is then found by

$$M_k = M_n + nt_k \qquad (2.32)$$

Once the mean anomaly is obtained, Kepler's equation can be iteratively solved.

$$M_k = E_k - esinE_k \qquad (2.33)$$

The true anomaly, v_k, is then calculated from Equations (2.25) and (2.26). Using the true anomaly, v_k, the argument of latitude can be determined:

$$\phi_k = v_k + \omega \qquad (2.34)$$

By inserting the argument of latitude into the following equations, the ephemeris correction terms for latitude, radius and inclination can be solved.

$$\delta u_k = C_{us}sin2\phi_k + C_{uc}cos2\phi_k \qquad (2.35)$$

$$\delta r_k = C_{rc}cos2\phi_k + C_{rs}sin2\phi_k \qquad (2.36)$$

$$\delta u_k = C_{ic}cos2\phi_k + C_{is}sin2\phi_k \qquad (2.37)$$

The corrections are then combined with the parameters at the satellite reference time to yield the corrected latitude, radius, and inclination.

$$u_k = \phi_k + \delta u_k \qquad (2.38)$$

$$r_k = A(1 - ecosE_k) + \delta r_k \qquad (2.39)$$

$$i_k = i_0 + \delta i_k \qquad (2.40)$$

The corrected latitude, radius, and inclination are then used to determine the satellite's position in the orbital plane:

$$x'_k = r_k cosu_k \qquad (2.41)$$

$$y'_k = r_k sinu_k \qquad (2.42)$$

The corrected longitude of the ascending node is found from:

$$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)t_k - \dot{\Omega}_e t_0 \qquad (2.43)$$

where Ω_0 and $\dot{\Omega}$ are defined in Table 2.1 and $\dot{\Omega}_e = 7.292115x10^{-5}rad/sec$, the WGS 84 value of the earth's rotation rate.

Using the orbital plane positions, the ECEF positions of the satellites can be computed using:

$$x_k = x'_k \cos\Omega_k - y'_k \cos i_k \sin\Omega_k \tag{2.44}$$

$$y_k = x'_k \sin\Omega_k - y'_k \cos i_k \cos\Omega_k \tag{2.45}$$

$$z_k = y'_k \sin i_k \tag{2.46}$$

The ECEF frame used in the above equations has the x direction in the true equatorial plane in the direction of the Greenwich meridian and the z axis along the true earth spin axis, positive in the northern hemisphere. This frame is then transformed to coincide with the Litton ECEF frame. The Litton ECEF frame has the z axis passing through the Greenwich meridian and the y axis along the true earth spin axis. (Refer to Figure 2.1) Since the LN-94 also provides information in terms of feet, the ECEF position was converted from metric to English units.

2.6 GPS Position Using Batch Least Squares

Chapter IV of this thesis will compare the results of the integrated GPS/INS system to the GPS-only navigation solution. For the calculation of position using GPS data only, this research uses batch least squares in determining the user position. Although this batch least squares method is rather slow computationally in comparison to the recursive Kalman filter, it is adequate since the platform is stationary and all calculations are done in a post-processing mode. The algorithm to perform the batch least squares is much simpler than the EKF and provides sufficient results (6). The following section describes the least squares theory in the context of solving only the GPS measurement equations to find a navigation position solution (21).
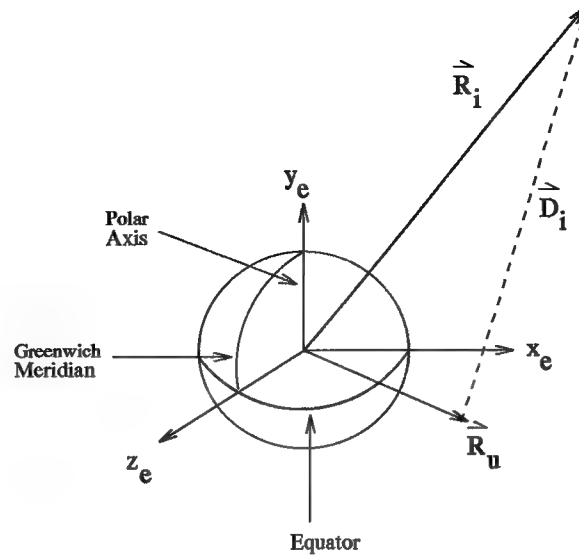
Figure 2.1 GPS Satellite and User ECEF Positions

Figure 2.1 shows the Litton ECEF reference frame. The user's position is indicated by the variable, $\vec{R}_u$, while the ECEF coordinates of the satellite are $\vec{R}_i$. The line-of-sight distance vector from the user to the satellite is shown as $\vec{D}_i$.

Let the user's position vector, $\vec{R}_u$, and the satellite's position, $\vec{R}_i$, be broken into the following components: x_u, y_u, z_u and x_i, y_i, z_i respectively, coordinatized in the Litton ECEF frame. The subscript i indicates which satellite is being tracked; therefore, there is one equation for each satellite. The distance variable, T, is the unknown user clock bias multiplied by the speed of light. The line-of-sight distance equations from the user to the satellite are given by

$$D_i = \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2 + (z_u - z_i)^2} + T \qquad (2.47)$$

In this equation there are four unknowns (three user positions and time); thus at least four equations are required corresponding to the four satellites in order to find a three-dimensional solution. However, these equations, are nonlinear and are very difficult to solve explicitly. An alternative is to linearize these equations about a nominal point. The initial nominal point is actually one's best guess. Let the nominal point be described as $x_n, y_n, z_n, T_n, D_{ni}$. Also define the corrections as Δx,

2-13

Δy, Δz, ΔT, and ΔD_i so that the true position is x=x_n+Δx, y=y_n+Δy, z=z_n+Δz, T=T_n+ΔT, and D_i=D_{ni}+ΔD_i. The nominal range to the satellite is

$$D_{ni} = \sqrt{(x_n - x_i)^2 + (y_n - y_i)^2 + (z_n - z_i)^2} + T_n \qquad (2.48)$$

The actual range can then be written as

$$D_i = D_{ni} + \Delta D_i = \sqrt{(x_n + \Delta x - x_i)^2 + (y_n + \Delta y - y_i)^2 + (z_n + \Delta z - z_i)^2} + T_n + \Delta T \qquad (2.49)$$

Linearizing this equation by using a first order Taylor series approximation, the equation becomes

$$\Delta D_i = \frac{(x_n - x_i)\Delta x}{D_{ni} - T_n} + \frac{(y_n - y_i)\Delta y}{D_{ni} - T_n} + \frac{(z_n - z_i)\Delta z}{D_{ni} - T_n} + \Delta T \qquad (2.50)$$

In this thesis, i (the number of satellites) will range from four to eight. Recall, the minimum number of satellites to solve a three dimensional problem is four. Equation (2.50) can be written in state-space form as

$$\begin{bmatrix} \Delta D_1 \\ \Delta D_2 \\ \Delta D_3 \\ \Delta D_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta T \end{bmatrix} \qquad (2.51)$$

where, $\alpha_{i1} = \frac{x_n - x_i}{D_{ni} - T_n}$, $\alpha_{i2} = \frac{y_n - y_i}{D_{ni} - T_n}$, $\alpha_{i3} = \frac{z_n - z_i}{D_{ni} - T_n}$ and i is the satellite number. A least squares solution to the equation can be found by iterating the equation until the error states get acceptably close to zero, producing the user's position when added to the nominal.

$$\hat{x} = (A^T A)^{-1} A^T \vec{y} \qquad (2.52)$$

Equations (2.51) and (2.52) are related by

$$\hat{x} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta T \end{bmatrix} \qquad (2.53)$$

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \qquad (2.54)$$

and

$$\vec{y} = \begin{bmatrix} \Delta D_1 \\ \Delta D_2 \\ \Delta D_3 \\ \Delta D_4 \\ \vdots \end{bmatrix} \qquad (2.55)$$

The following discussion presents the derivation of Equation (2.52).

In the GPS-data-only solution, there is a possibility of eight equations with the four previously specified unknowns. To illustrate the least squares method, consider the equation $A\vec{x} = b$, shown in Figure 2.2. If A provides more equations than unknowns, then b almost certainly will not fall in the column space $\mathcal{R}(A)$. In other words, the number of observations is larger than the number of unknowns, so it is expected that $A\vec{x} = b$ will be inconsistent. There will not exist a choice of $\vec{x}$ that perfectly fits the data b. When determining a solution, the least squares method minimizes the average error in a series of equations. Figure 2.2 shows geometrically the least squares solution.

In order to solve the equation $A\vec{x} = b$, b must be projected onto the subspace. The dotted line in Figure 2.2 is the error vector, $b - A\vec{x}$. This error vector is perpendicular to the subspace (23).
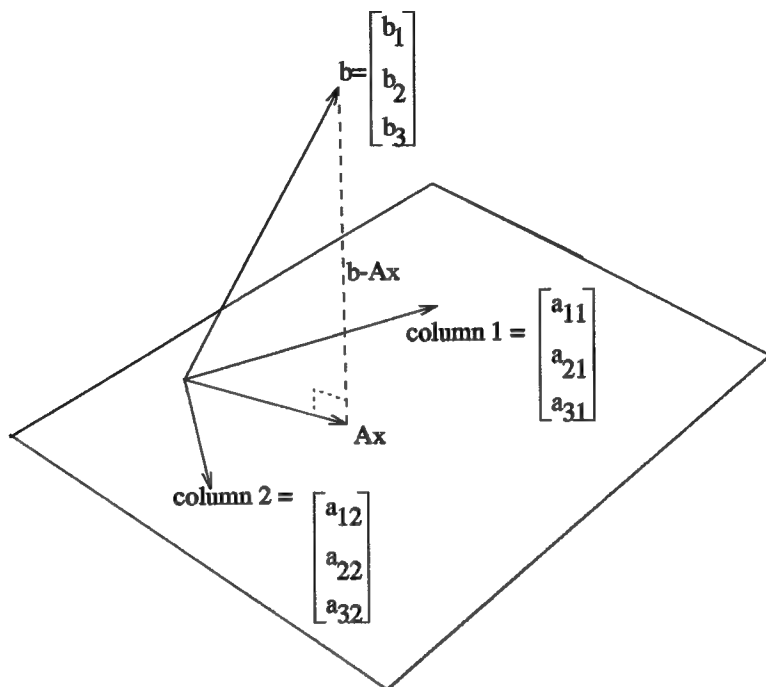
Figure 2.2   Least Squares Solution

The least square error is $E = \|A\vec{x} - b\|$ where $\vec{x}$ minimizes the distance from b to the point $A\vec{x}$ in the column space. $A\vec{x}$ is the combination of the columns with coefficients $x_1, ..., x_n$. Searching for the least squares solution, which minimizes E, is the same as locating the point $p = A\vec{x}$ that is closer to b than any other point in the column space. In reference to Fig 2.2, the error vector $b - A\vec{x}$ must be perpendicular to that space. The calculation of $\vec{x}$ and the projection $A\vec{x}$ can be accomplished in two ways (23).

1. The vectors perpendicular to the column space lie in the left nullspace. Thus the error vector $b - A\vec{x}$ must be in the nullspace of A^T:

$$A^T(b - A\vec{x}) = 0 \text{ or } A^T A\vec{x} = A^T b \tag{2.56}$$

2. The error vector must be perpendicular to every column of A:

$$\begin{bmatrix} a_1^T \\ \cdot \\ a_n^T \end{bmatrix} \begin{bmatrix} b - A\vec{x} \end{bmatrix} = 0. \tag{2.57}$$

Equations (2.56) and (2.57) are equivalent to Equation (2.52). Thus, when given GPS position information only, a batch least squares method is used. The GPS position and time estimates are then found by iterating Equation (2.52) until the errors get very close to zero.

2.7 Chapter Summary

This chapter reviewed extended Kalman filtering theory. The EKF will be the primary integration tool used in combining the GPS pseudorange measurements and the INS navigation solution. The discussion then moved to filter order reduction and filter tuning. These subjects are critical to this thesis, since every INS model used in both the system truth and filter models is reduced from the Litton LN-93 truth model. The discussion then presented the equations used in determining the GPS satellite ECEF positions. The chapter concluded with a presentation of batch least squares theory, which will be used in the determination of the GPS-only solution.

III. Navigation Models

This chapter begins with a discussion of the computer simulation model and continues with a presentation of the GPS/INS hardware integration. The chapter then describes the INS and GPS models used within the extended Kalman filter in both the simulation and hardware integration. Both the system truth and filter models are given, as well as the measurement equations used.

3.1 Computer Simulation

The goal of the computer simulation is to duplicate the actual GPS/INS hardware integration. This simulation is accomplished with the aid of system truth models which characterize, as closely as possible, the errors of the GPS and INS, trajectory generators which provide a flight profile, and a satellite orbit generator which provides the Earth-Centered-Earth-Fixed (ECEF) coordinates of the satellites. Of course these models are not perfect, but hopefully, they will provide the required precision in order to accurately determine the quality of the navigation system design.

Figure 3.1 breaks down the computer simulation into its major components and illustrates the signal flow. This figure is referred to throughout the chapter. Keep in mind, the block diagrams used are actually a collection of MATLAB m-files written to provide the desired outputs.

The Flight Trajectory Generator transmits information pertaining to the vehicle's position and orientation. Specifically, the Flight Trajectory Generator provides vehicle position, velocity, acceleration, roll, pitch, yaw, and the rates of change of the roll, pitch, and yaw. In Figure 3.1, the output of the Flight Trajectory Generator is $\mathbf{X}$, the true position. For this research, the Flight Trajectory Generator was relatively simple since the platform was to remain stationary. For the stationary case, all variables of interest were zero except for the position coordinates of the LN-94 and the GPS antenna. The Flight Trajectory Generator provides the information to three places: (1) the INS system truth model to compute the INS simulated errors, (2) the summing junction
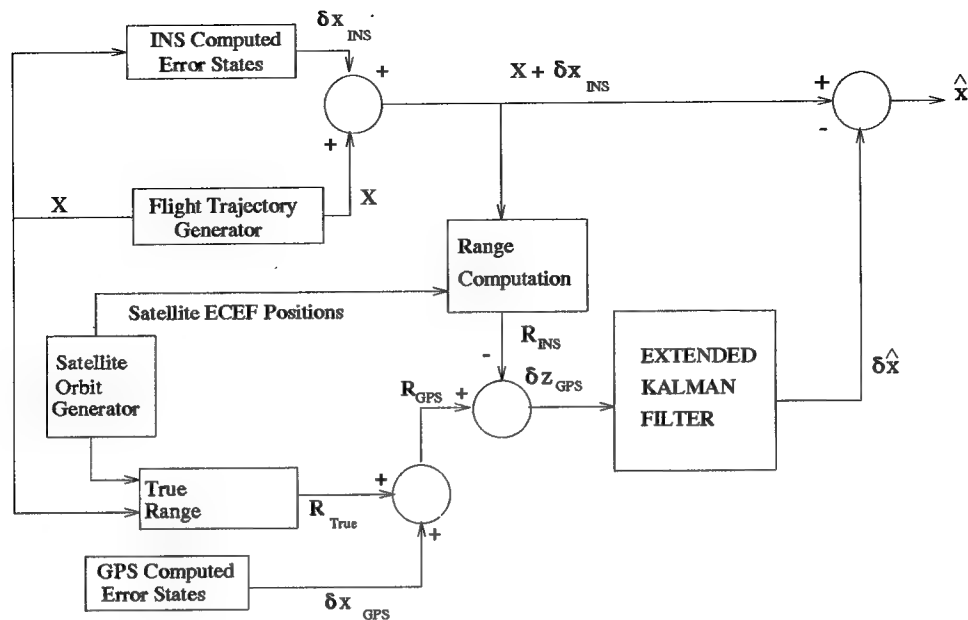
Figure 3.1   Computer Simulation Block Diagram for the Feedforward GPS/INS Integration

to the right of the INS system truth model block to provide simulated INS output, and (3) to the GPS summing junction to produce the simulated GPS receiver output.

The INS Computed Error States Block contains the 39-state INS truth model. The truth model will be presented in detail in Section 3.3.2. The output of the INS Computed Error States Block, δx_{INS}, is summed with the true position, $\mathbf{X}$, to produce the simulated INS output.

The Satellite Orbit Generator calculates the ECEF positions of four satellites. This subroutine is capable of providing nearly any GDOP; however, for the simulation the satellite orbit trajectories were positioned to provide an average GDOP of 3.22. This particular value was selected because it is a representative number found when collecting real data from the XR-4PC and XR-5PC GPS receivers. The Satellite Orbit Generator supplies information to the GPS summing junction to compute the GPS pseudorange and to the INS Range Computation block in order to calculate the INS predicted pseudorange.

The True Range Block outputs R_{True}, the true range from the user to the satellite. The True Range Block transforms the Navigation Frame user position coordinates, $\mathbf{X}$, from the Flight

Trajectory Generator to ECEF coordinates. The satellite ECEF position is then differenced with the user position to determine the true range from the satellite to the user, R_{True}. The output of the True Range Block is combined with the errors from the GPS truth model located in the GPS Computed Error States to produce the GPS pseudorange, R_{GPS}.

The Range Computation Block receives the simulated INS output and the satellite ECEF positions. The INS output, $\mathbf{X} + \delta\mathbf{x}_{INS}$ is converted to the ECEF coordinate frame in the Range Computation Block. Then the satellite ECEF position is subtracted from the user ECEF position to determine the INS computed pseudorange, R_{INS}.

The purpose of the summing junction prior to the EKF block is to difference the INS computed pseudorange from the GPS pseudorange. The output of this summing junction is the sensor measurement error, δz_{GPS}, which is provided to the Extended Kalman Filter.

The EKF block uses the sensor measurement error to compute the position, velocity, acceleration, and attitude errors. Extended Kalman Filter theory is given in Chapter II. In the feedforward implementation, these filter computed errors are subtracted from the INS output to provide a best estimate of the user's navigation solution. With the feedback implementation, shown in Figure 3.2, the EKF output is supplied directly back to the INS in order to correct the position, velocity, and acceleration states before the INS computations are conducted. In the feedback configuration, the INS output is the best estimate of the navigation solution.

3.2   Hardware Integration

This section will relate the actual hardware integration to the blocks shown in Figure 3.3. Once again, the filter models and EKF equations are implemented in MATLAB m-files. The data was collected and stored in data files from the GPS receivers and LN-94 INS. This information was then fed to the EKF and post-processed in a *feedforward* implementation.
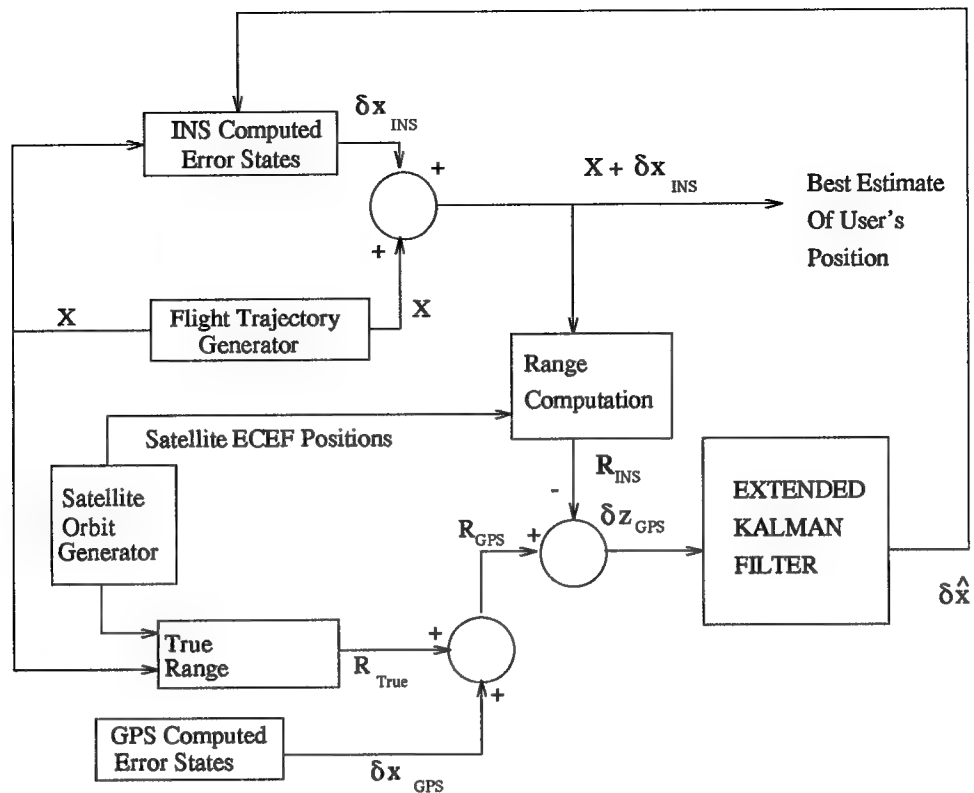
Figure 3.2   Computer Simulation Block Diagram for the Feedback GPS/INS Integration

The GPS data collected consisted of satellite ephemeris data and pseudorange measurements, R_{GPS}. An algorithm was written to calculate the satellite ECEF positions from the satellite ephemeris data as described in Section 2.5. These satellite positions were then provided to the Range Computation Block. The LN-94 position, velocity, acceleration, wander angle, and attitude information was collected from the 1553 data bus. The LN-94 data was used in conjunction with the satellite ECEF positions in the INS Range Computation Block to compute the INS-predicted pseudorange, R_{INS}.

The INS-computed pseudorange, R_{INS}, and the GPS pseudoranges, R_{GPS}, are then differenced at the junction prior to the EKF to produce the error in the sensor measurement. The EKF then uses these measurements to predict the errors in the navigation solution. The EKF error estimates are then subtracted from the INS output in the feedforward configuration, or fed back to the INS in the feedback implementation, to provide a navigation solution.
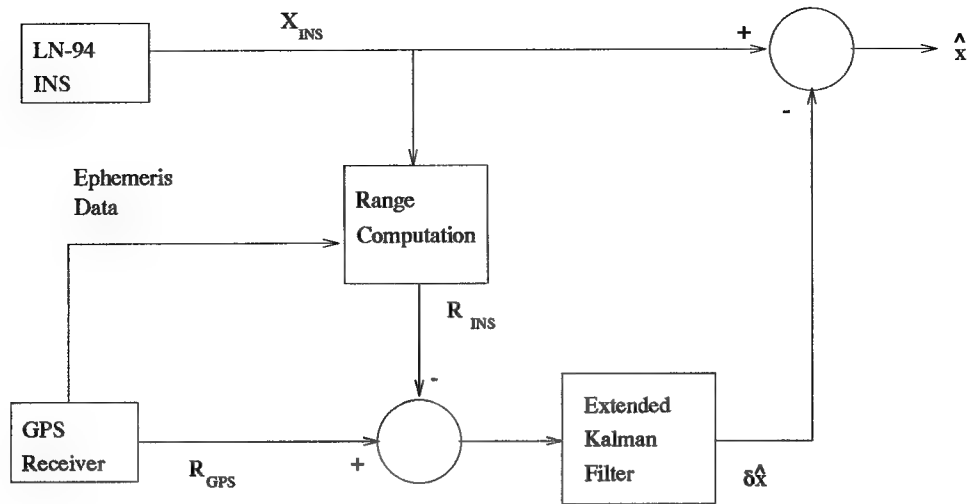
Figure 3.3   Feedforward GPS/INS Hardware Integration

3.3   INS Models

The system truth and filter models described in this section are derived from the 93-state Litton model (8). This section begins with a description of the revised 93-state model. The 39-state system truth model used in the computer simulation is then presented. The section concludes with the 11-state filter model used in both the computer simulation and the hardware integration.

3.3.1   The 93-State LN-93 Error Model.   The 93-state INS system truth model used in this research is a revision of the original Litton LN-93 state INS model. The modifications to the Litton model were made as a result of research done by the Wright Labs Avionics Directorate Research Group and past AFIT theses' results (18). The Litton model is composed of 93 states; these 93 error states are further broken down into six categories. Appendix A contains a table of the states and their description. The six major categories of these 93 states are given below:

$$\delta \mathbf{x} = \left[\delta \mathbf{x}_1^T \, \delta \mathbf{x}_2^T \, \delta \mathbf{x}_3^T \, \delta \mathbf{x}_4^T \, \delta \mathbf{x}_5^T \, \delta \mathbf{x}_6^T \right]^T \tag{3.1}$$

where:

$\delta \mathbf{x}_1$ contains the first 13 states which are position, velocity, attitude, and vertical channel errors. These states are classified

3-5

as "general" errors corresponding to standard Pinson error model states and states associated with barometric altimeter aiding of the vertical channel.

$\delta \mathbf{x_2}$ represents the "trend" states and are modeled as first-order Markov processes in the system truth model. This category is composed of the 16 gyro, accelerometer and baro-altimeter errors.

$\delta \mathbf{x_3}$ consists of gyro bias errors, which are modeled as random constants.

$\delta \mathbf{x_4}$ is also modeled as random constants and is made up of the accelerometer bias errors.

$\delta \mathbf{x_5}$ is a set of first order Markov processes and is composed of the six accelerometer and gyro initial thermal transients.

$\delta \mathbf{x_6}$ is composed of the 18 gyro compliance error states. These states are modeled as biases in the system truth model.

The 93-state Litton model state space differential equation is given by:

$$
\begin{bmatrix} \delta \dot{\mathbf{x}}_1 \\ \delta \dot{\mathbf{x}}_2 \\ \delta \dot{\mathbf{x}}_3 \\ \delta \dot{\mathbf{x}}_4 \\ \delta \dot{\mathbf{x}}_5 \\ \delta \dot{\mathbf{x}}_6 \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{F}_{13} & \mathbf{F}_{14} & \mathbf{F}_{15} & \mathbf{F}_{16} \\ 0 & \mathbf{F}_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{F}_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_1 \\ \delta \mathbf{x}_2 \\ \delta \mathbf{x}_3 \\ \delta \mathbf{x}_4 \\ \delta \mathbf{x}_5 \\ \delta \mathbf{x}_6 \end{bmatrix} + \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.2}
$$

This information was taken from the Litton LN-93 Error Budget (8). This truth model is the most detailed simulation model available for the LN-93.

3.3.2 The 39-State LN-93 Error Model. The 39-state LN-93 model presented in this section is used as the system truth model in this research. Although the 93-state LN-93 model is the most accurate characterization available, the high degree of dimensionality makes it computationally burdensome. Previous AFIT research conducted by Lewantowicz (9) and Negast (20) have demonstrated that the 39-state is a sufficient substitute for the 93-state model without a significant loss of accuracy. The 39-state model eliminates all states that do not produce significant errors. The state space equation, (3.2), is reduced to:

$$
\begin{bmatrix}
\delta \dot{\mathbf{x}}_1 \\
\delta \dot{\mathbf{x}}_2 \\
\delta \dot{\mathbf{x}}_3 \\
\delta \dot{\mathbf{x}}_4
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{F}_{13} & \mathbf{F}_{14} \\
0 & \mathbf{F}_{22} & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta \mathbf{x}_1 \\
\delta \mathbf{x}_2 \\
\delta \mathbf{x}_3 \\
\delta \mathbf{x}_4
\end{bmatrix}
+
\begin{bmatrix}
\mathbf{w}_1 \\
\mathbf{w}_2 \\
0 \\
0
\end{bmatrix}
\tag{3.3}
$$

The four components of the 39-state vector do not directly correlate to the first four categories of the 93-state model as described in Section 3.3.1. The tables in Appendix A.7 give a complete description of the states.

3.3.3   The 11-State LN-93 Error Model.   The 11-state filter model is used in both the hardware integration and the computer simulation Extended Kalman Filters. The 11-state filter consists of the first 11 states taken from the 93-state LN-93 model. This particular model has not been employed in past AFIT research. Previous AFIT research has used an 11-state filter; however, the last state in the filter model was state 23 of the LN-93 truth model (18, 20). AFIT research found that state 23, *the total baro-altimeter correlated error*, provided the best results, since the barometric altimeter was used as a sensor measurement. In this research, the barometric altimeter is used to stabilize the LN-94's vertical channel, but is not taken as a separate measurement for the EKF. Through experimental research, state 11, *the error in the lagged inertial altitude*, produced significantly better altitude estimates than state 23. Appendix C provides a description of the research and a comparison of the results leading to this filter model change.

3.4   GPS Models

There are four GPS models described in this section: the 30-state system truth, the 22-state, the 10-state, and the 2-state filter models. This thesis will refer to these models by their minimum number of states (assuming that four satellites are being used); however, in reality the model may have more states depending upon the number of satellites being used for a navigation solution. For instance, the 30-state model is actually a 58-state model when eight satellites are being tracked. In addition this section also presents the pseudorange measurement equations; these equations provide the basis for the development of the observation matrix used in the EKF.

3.4.1 30-State GPS Model.   The 30-state GPS system truth model is comprised of five types of error sources. By far, the most significant of these errors are the user clock bias and the user clock drift. The first two states model these errors as follows:

$$
\begin{bmatrix} \delta \dot{\hat{R}}_{clku} \\ \delta \dot{\hat{D}}_{clku} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta R_{clku} \\ \delta D_{clku} \end{bmatrix}
\tag{3.4}
$$

where:

$$
\delta \dot{\hat{R}}_{clku} = \text{range equivalent of user clock bias}
$$

$$
\delta \dot{\hat{D}}_{clku} = \text{velocity equivalent of user clock drift}
$$

The initial state estimates and covariances were taken from past AFIT theses:

$$
\begin{bmatrix} \delta \dot{R}(t_0) \\ \delta \dot{D}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\tag{3.5}
$$

and

$$
\mathbf{P}_{\delta R_{clku}, \delta D_{clku}}(t_0) = \begin{bmatrix} 9.0x10^{14} ft^2 & 0 \\ 0 & 9.0x10^{10} ft^2/sec^2 \end{bmatrix}
\tag{3.6}
$$

There remain seven error states for each of the four satellites. These seven errors are composed of the code loop, tropospheric, ionospheric, satellite clock, and three line-of-sight errors. The code loop, tropospheric, and ionospheric errors are modeled as first-order Markov processes. The remaining errors are modeled as random constants (18). The state space representation is shown below:

3-8

$$
\begin{bmatrix} \delta \dot{R}_{cl} \\ \delta \dot{R}_{trop} \\ \delta \dot{R}_{ion} \\ \delta \dot{R}_{sclk} \\ \delta \dot{x}_i \\ \delta \dot{y}_i \\ \delta \dot{z}_i \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{500} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{1500} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta R_{cl} \\ \delta R_{trop} \\ \delta R_{ion} \\ \delta R_{sclk} \\ \delta x_i \\ \delta y_i \\ \delta z_i \end{bmatrix} + \begin{bmatrix} w_{cl}(t) \\ w_{trop}(t) \\ w_{ion}(t) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\tag{3.7}
$$

where:

$$
\mathbf{w}_{GPS}(t) = \begin{bmatrix} w_{cl} \\ w_{trop} \\ w_{ion} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\tag{3.8}
$$

The initial covariance for the states is given by:

$$
\mathbf{P}_{GPS}(t_0) = \begin{bmatrix} 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25 \end{bmatrix} ft^2
\tag{3.9}
$$

The mean values and strengths of the dynamic driving noise are given by:

$$
\mathbf{E}\{\mathbf{w}_{GPS}(t)\} = \mathbf{0}
\tag{3.10}
$$

$$\mathbf{E}\{\mathbf{w}_{GPS}(t)\mathbf{w}_{GPS}^{T}(t+\tau)\} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.004 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} ft^{2}/sec \cdot \delta(\tau) \qquad (3.11)$$

Appendix A.10 presents the state numbers and their descriptions. The noises used are presented in Chapter IV.

3.4.2   22-State GPS Model.   The 22-state model is very similar to the 30-state model. The difference between the two models is that the three directional error states in the 30-state model are combined into one line-of-sight error. The 22-state model is an experiment by this researcher. The model is a melding of the 10-state and 30-state models. Since the pseudorange comprises only one line-of-sight measurement, it seems reasonable that all of the satellite line-of-sight error can be represented in one state. Of course, the dynamics driving noise will be somewhat different, and the error variance used is just the sum of the three directional variances given in the 30-state model. The complete description of the 22-state model is presented in Appendix A.11.

3.4.3   10-State GPS Model.   The basis for this model was taken from the MSOFE GPS/INS integration sample problem (3). The model has not been used in past AFIT theses, but was used in the development of the MATLAB m-files used in this thesis (7). The model contains the receiver-related errors, such as user clock drift and receiver noises in the first two states. The remaining errors are the satellite-related (satellite clock and orbital ephemeris) and path-related (tropospheric and ionospheric delays) errors, which are lumped into the last eight states. The last eight states (two per each of four satellites) are simply pseudorange and pseudorange-rate error.

3.4.4   2-State GPS Model.   The 2-state filter is rather small when compared to the 30-state system truth model. A model reduction of this order is possible because of the size of the user clock bias and drift errors in comparison to the remaining states. The filter model is shown below:

3-10

$$\begin{bmatrix} \delta \dot{R}_{clku} \\ \delta \dot{D}_{clku} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta R_{clku} \\ \delta D_{clku} \end{bmatrix} + \begin{bmatrix} w_{Rclk} \\ w_{Rclk} \end{bmatrix} \tag{3.12}$$

Due to the filter order reduction, the dynamics driving noise, $\mathbf{Q}$, must be increased to compensate for the added uncertainty in the model. Chapter 4 compares the various values of $\mathbf{Q}$ for the different models.

3.4.5 GPS Measurement Model.   Since four satellites are used in the computer simulation, there are four GPS measurement updates. The hardware integration uses up to eight satellites in view; therefore, the number of measurements depends upon the number of satellites being tracked by the GPS receiver. A subroutine was written to set the observation matrix to account for all available satellites. Mosle (18) was used as a reference for the GPS measurement model development. The pseudorange measurements received by the GPS receiver are differenced with the INS computed pseudorange to produce a difference measurement:

$$\delta z_{GPS} = R_{INS} - R_{GPS} \tag{3.13}$$

The pseudorange, R_{GPS}, is the sum of the true range from the user to the satellite plus the errors.

$$R_{GPS} = R_t + \delta R_{cl} + \delta R_{trop} + \delta R_{ion} + \delta R_{uclk} - \delta R_{sclk} - v \tag{3.14}$$

where:

R_{GPS} = GPS pseudorange measurement, from satellite to user

R_t = true range, from satellite to user

δR_{cl} = range error due to code loop error

δR_{trop} = range error due to tropospheric delay

δR_{ion} = range error due to ionospheric delay

δR_{uclk} = range error due to user clock

δR_{sclk} = range error due to satellite clock

v = zero-mean white Gaussian measurement noise

The INS-computed pseudorange measurement is found by differencing the satellite ECEF positions from the ECEF position provided by the INS:

$$R_{INS} = |\mathbf{X}_u - \mathbf{X}_s| = \left\| \begin{bmatrix} x_u \\ y_u \\ z_u \end{bmatrix}^e - \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}^e \right\| \tag{3.15}$$

The above equation can be written as:

$$R_{INS} = \sqrt{(x_u - x_s)^2 + (y_u - y_s)^2 + (z_u - z_s)^2} \tag{3.16}$$

Equation (3.16) can be approximated and rewritten in terms of the true range and a truncated first-order Taylor series with perturbations $\delta\mathbf{x_u}$ and $\delta\mathbf{x_s}$ representing the errors in $\mathbf{X_u}$ and $\mathbf{X_s}$:

$$R_{INS} = R_t + \left.\frac{\partial R_{INS}(\mathbf{X}_s, \mathbf{X}_u)}{\partial \mathbf{X}_u}\right|_{(\mathbf{X}_s,\mathbf{X}_u)_{nom}} \cdot \delta\mathbf{X}_u + \left.\frac{\partial R_{INS}(\mathbf{X}_s, \mathbf{X}_u)}{\partial \mathbf{X}_s}\right|_{(\mathbf{X}_s,\mathbf{X}_u)_{nom}} \cdot \delta\mathbf{X}_s \tag{3.17}$$

The solution is found by substituting Equation (3.16) into Equation (3.17) and evaluating the partial derivatives:

$$\begin{aligned} R_{INS} = R_t &- \left[\frac{x_s - x_u}{|R_{INS}|}\right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|}\right] \cdot \delta y_u - \left[\frac{z_s - z_u}{|R_{INS}|}\right] \cdot \delta x_u \\ &+ \left[\frac{x_s - x_u}{|R_{INS}|}\right] \cdot \delta x_s + \left[\frac{y_s - y_u}{|R_{INS}|}\right] \cdot \delta y_s + \left[\frac{z_s - z_u}{|R_{INS}|}\right] \cdot \delta z_s \end{aligned} \tag{3.18}$$

The GPS pseudorange truth model difference measurement is given by:

$$\begin{aligned} \delta z_{GPS(true)} = R_{INS} - R_{GPS} = &- \left[\frac{x_s - x_u}{|R_{INS}|}\right] \cdot \delta x_s - \left[\frac{y_s - x_u}{|R_{INS}|}\right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|}\right] \cdot \delta y_u \\ &- \left[\frac{z_s - z_u}{|R_{INS}|}\right] \cdot \delta z_u + \left[\frac{x_s - x_u}{|R_{INS}|}\right] \cdot \delta x_s + \left[\frac{y_s - y_u}{|R_{INS}|}\right] \cdot \delta y_s + \left[\frac{z_s - z_u}{|R_{INS}|}\right] \cdot \delta z_s - \partial R_{cl} \\ &- \partial R_{trop} + \partial R_{sclk} - \partial R_{uclk} - v \end{aligned} \tag{3.19}$$

The measurement noise present in the GPS difference measurement truth model represents sensor noise and modeling uncertainties. The statistical characterization of these noises are presented in Chapter IV.

The 2-state filter model GPS pseudorange difference measurement is written as:

$$\delta z_{GPS} = R_{INS} - R_{GPS} = -\left[\frac{x_s - x_u}{|R_{INS}|}\right] \cdot \delta x_u - \left[\frac{y_s - y_u}{|R_{INS}|}\right] \cdot \delta y_u - \left[\frac{z_s - z_u}{|R_{INS}|}\right] \cdot \delta z_u$$
$$-\partial R_{uclk} - v \qquad (3.20)$$

Equations (3.19) and (3.20) differ in the omission of the system truth model atmospheric errors, satellite clock errors, and satellite position errors. The filter measurement noise variance, R, is tuned to compensate for the reduction in order from the truth model. The measurement noise statistics are given in Chapter 4 and compared to those used in the truth model.

3.5   Chapter Summary

This chapter has presented a general overview of the computer simulation employed and its relation to the actual hardware integration. In addition, the INS and GPS truth and filter models were presented. The system truth models were used in the computer simulation to represent the actual LN-94 and GPS receivers. The filter models were used in both the hardware integration and the computer simulation. The chapter concluded with the measurement equations necessary to update the EKF .

a

IV. Results and Analysis

This chapter presents the results of the GPS/INS integration research. The chapter is divided into two major sections: computer simulation and hardware integration. The chapter begins with the computer simulation results for a 69-state filter and the 13-state filter using the 69-state system truth model. The filters were tuned with no knowledge of the hardware integration dynamic driving noise strength tuning values. After the hardware integration, the actual dynamic driving noise values were inserted into the simulation to demonstrate differences between the real world and simulation. A comparison of the actual tuning versus the simulation was accomplished. The discussion then moves to the collection and analysis of LN-94 and GPS receiver data. This data was used to illustrate the improvement in navigation positioning accuracy via the Kalman filter. The next step was to present the results of combining the data with an EKF in a tightly-coupled integration. Five different GPS models were used in the filter to demonstrate variations in performance. The measurement update rate was then investigated to determine how much improvement in performance can be obtained by increasing the frequency of measurement updates. The final section of the chapter discusses the subject of feedback or closed-loop integration. The results of long term runs demonstrate when resetting the INS in a feedback implementation would be advantageous over a feedforward approach.

4.1 Tuning

Although there were several surprises, nearly every result in this research provides an outcome that one would expect, given the theory. Yet, these results are very dependent on tuning. It was an eye-opening experience for this researcher to learn the range of values that could be obtained by changing tuning of the dynamic driving noises and the sensor measurement noises. The EKF is very versatile and can be tuned for nearly any situation. The problem is that its performance may be excellent in the environment for which it is tuned, but it may be rather lacking if conditions change, such as a failure.

Tuning for the computer simulation was very conservative in order to meet the criterion established in Section 2.4. The tilt error states were very difficult to tune. The tuning of the tilt error states was dependent on the tuning of nearly every other error state. In other words, tuning

of the clock bias error state or the latitude error state changed the performance of the tilt error states. All other error states could be tuned in a systematic fashion.

Since this thesis strives to provide an unbiased report on the performance of the different filters, a groundwork for tuning was established. Once satisfactory filter performance was attained, the dynamic driving noise strengths for the INS error states were not changed. These values are shown later in Section 4.2, Table 4.1. Only the dynamic driving noises associated with the GPS states and the sensor measurement noises were altered to improve performance. Interestingly, the filter was remarkably robust and could be tuned for many different situations; however, there were trade-offs. For instance, if a long duration run was accomplished, then shorter term performance was sacrificed. In an extreme case, the filter could be tuned so that the models were discounted and essentially no more than the GPS solution was provided. In general, the shorter the tuning interval, the better the EKF can predict the navigation solution errors. Nonlinearities unmodeled by the linearized filter model become more significant as time passes. The superior short-term accuracy is because the INS modeling errors have not grown as large in the initial stages.

The measurement update rate also had an effect on the tuning. Accuracy in filter estimates increased as the measurement update frequency increased. The accuracy increases only if the tuning values are changed to reflect the altered update rate. When the frequency of measurement updates increased, the filter propagated for a shorter period of time. Rather than decreasing the dynamic noise strengths on the model, the sensor measurement noise was increased. This method made tuning of the filter much easier. Only one value was varied versus changing the Q on each error state. The results of changing the measurement update frequencies is developed further in Section 4.4.4.

4.2   Computer Simulation

This section presents the results of the integrated GPS/INS computer simulations. Two different sets of simulations were run; these simulations are categorized by the way they were tuned. The first section describes a GPS/INS integration simulation tuned with no knowledge of the actual hardware integration tuning results. The criterion used in tuning the filter is given in Chapter II. The second section uses the real data tuning values used in the actual hardware integration. Both categories of simulation analyzed the 69-state and 13-state filters. The truth model for the computer

simulations is composed of the 39-state LN-93 truth model given in Appendix A.7 and the 30-state GPS model appearing in Appendix A.10, which will be referred to throughout the text as the 69-state model.

All of the simulations are for tightly-coupled feedforward GPS/INS integrations. The simulations were set up to represent the hardware integration as closely as possible for later comparison. The initial conditions were the exact surveyed coordinates of the laboratory used in aligning the INS and the initial covariances for the EKF are given in Chapter 3. The simulations were run successfully using GPS pseudorange measurements only. Unlike past AFIT research (10, 11, 18, 20, 22, 27), no velocity aiding from perfect Doppler radar measurements was needed. Also the simulation deleted the barometric altimeter measurement because an actual altimeter is not available in the laboratory. All of the simulations were conducted using an AFIT-developed computer routine of MATLAB m-files; see Appendix J. The resulting plots are from a Monte Carlo analyses consisting of 10 runs. The measurement sample period was 20 seconds, which is the actual data collection frequency used in the hardware integration. The 20 second interval used in GPS data collection was to prevent data files from becoming large and unmanageable.

4.2.1 Category I: Simulation Results. Simulations which were tuned with no knowledge of the actual hardware integration dynamic driving noise values are identified as Category I. Basically, the filter was tuned by adjusting the dynamic driving noise so that the mean ± one sigma of estimation error is contained within the envelope formed by the ± one filter-predicted sigmas. By using the dynamic driving noise, the filter could be tuned more precisely. The sensor measurement noise, R, has a general effect on the tuning of all error states and the initial covariances, P_0, only affect the initial transients. These values were established in past AFIT theses and were assumed applicable to this simulation (18, 20, 22, 27). Tables 4.1 through 4.4 provide the noise values and the plots are given in Appendices D and E.

4.2.1.1 69-State Filter Model. This simulation compares a 69-state system truth model to the 69-state filter. The 69-state filter used in the first GPS/INS computer simulation is nearly the same model as the system truth model. The 69-state model is composed of the 30-state GPS and the 39-state INS models. The only difference between the system truth and the filter models is the dynamics driving noise strength values. A slight amount of noise was added to all of
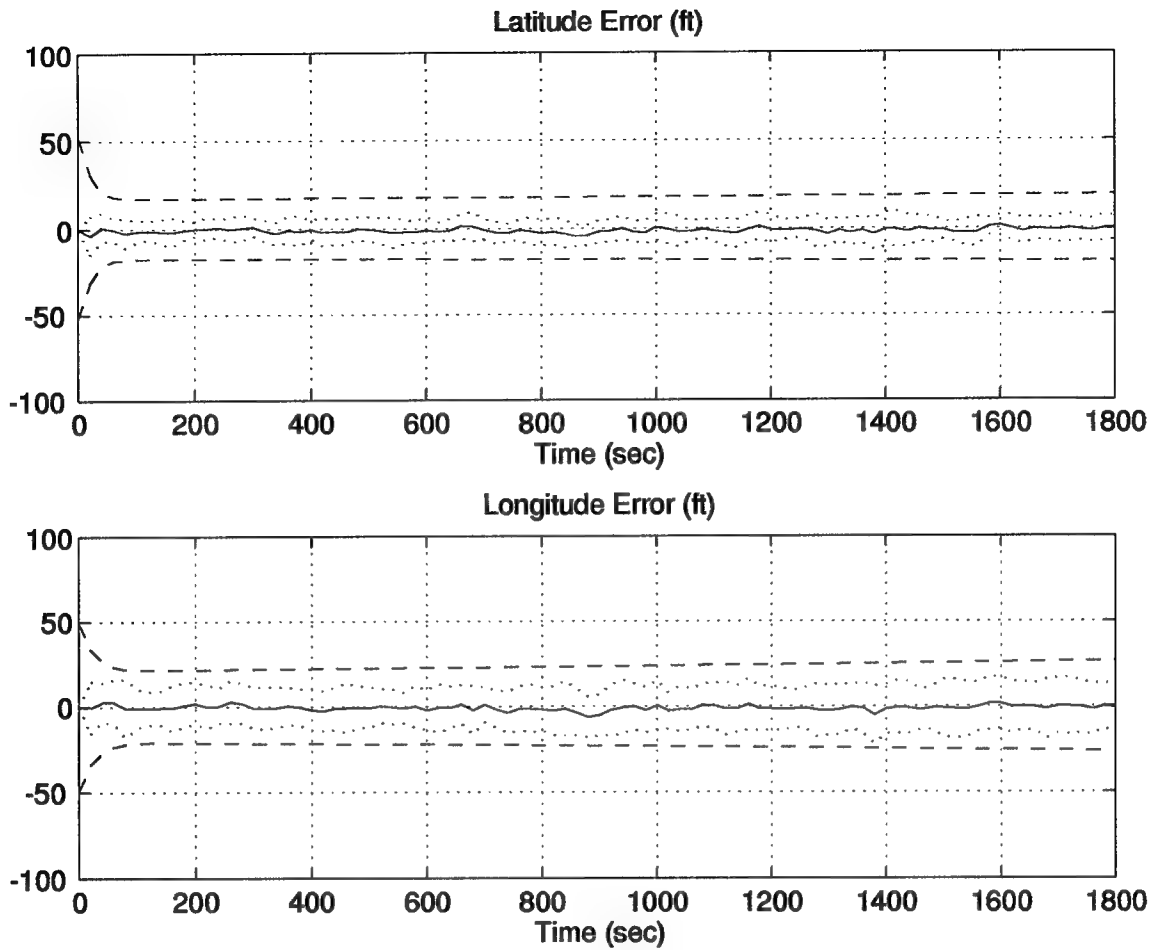
Figure 4.1   Category I: Position Errors for the 69-State Filter Computer Simulation

the error states in addition to the system truth noise, in order for the filter to follow the system truth model. The additional noise opened the bandwidth of the filter slightly to compensate for differences between the filter and system truth models caused by random number generation.

The purposes of using a filter of the same order as the system truth model in a computer simulation are for comparison to the hardware integration and to provide a benchmark to compare performances of all reduced-order filters. In the hardware integration, the 69-state filter was chosen to investigate the adequacy of the 69-state system truth model. The 69-state filter will demonstrate how well the model emulates the errors of the real world INS and GPS systems.

The latitude and longitude plots in Figure 4.1 show the position errors for the simulation. All of the 69-state filter model plots are given in Appendix D. In the plots, the solid line represents
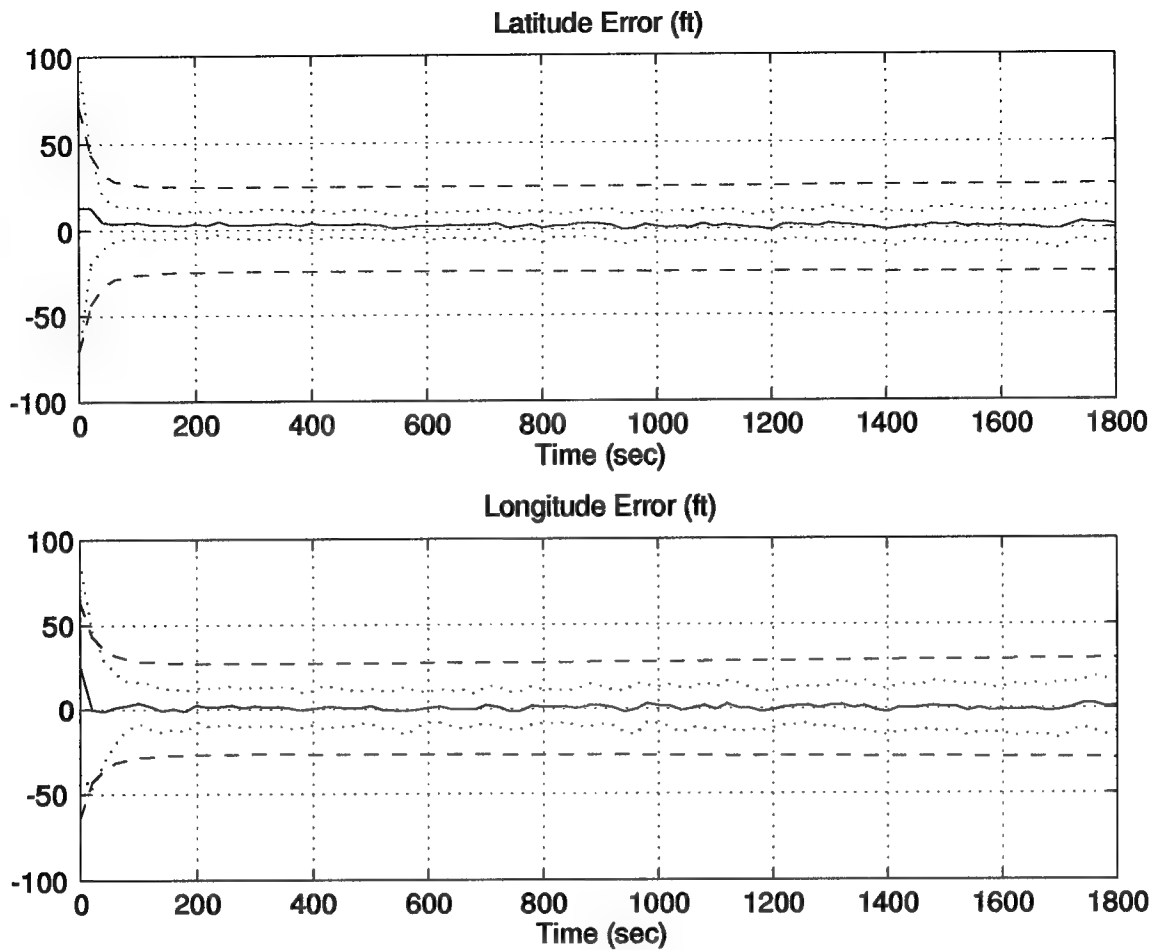
Figure 4.2   Category I: Position Errors for the 13-State Filter Computer Simulation

mean filter error, the dotted line is the actual mean error ± one sigma (one standard deviation), and the dashed line is the zero filter-predicted mean error ± one sigma. The filter was conservatively tuned in order to meet the criterion established in Section 2.4 and because of the difficulties posed by the tilt error states as discussed in Section 4.1.

4.2.1.2   13-State Filter Model.   The second Category I simulation used a 13-state filter model. This filter model is composed of the 11-state INS model and the two-state GPS model. Again, the 69-state model is used as the system truth model. Both of these models are described in Chapter III. The 13-state filter was chosen because of its probable use in a future real-time GPS/INS tightly-coupled integration.

4-5

As would be expected, the mean error ± one sigma and the zero ± one sigma filter-predicted errors are somewhat larger. This wider range of values represents greater error in the EKF estimates. The larger errors are because the number of filter states has been reduced from 69 to 13 states and the amount of uncertainty in the model is therefore increased.

The latitude and longitude plots are shown in Figure 4.2. All of the plots produced from the 13-state filter 10-run Monte Carlo analysis are shown in Appendix E.

4.2.2 Category II: Simulation Results. After the hardware integration was completed, the dynamic driving noise strength values from that integration were inserted back into the computer simulation filter models and a 10-run Monte Carlo analysis was run. The system truth model remained the same as in Category I, and the simulation initial conditions were unchanged. The only differences in the Category I and II simulations were the filter dynamic driving and sensor noises. A change in the filter tuning philosophy was required for the real data because the actual pseudoranges were much noisier than the computer simulation's representation. The EKF would rapidly diverge if the computer simulation sensor measurement noise strengths were used. The sensor measurement noises are given in Tables 4.2 and 4.4. Additional reasons for increasing the sensor measurement noise strengths were to compensate for user clock bias shifts (discussed in Appendix C.) and the substitution of satellites by the receiver. To explain the latter point, the GPS receiver is capable of receiving signals from eight different satellites. If another satellite appears in the receiver's field of view with a stronger signal, then the receiver will replace a previously used satellite with the more desirable one. By changing the sensor measurement noises in the real data scenario, the filter was tuned rather quickly and easily. When tuning with the dynamic driving noise strengths, each error state affected the model differently and tuning was tougher to accomplish. Although some dynamic driving noise strength tuning was accomplished, the sensor measurement noise strengths were used most often. In the use of real data, tuning was accomplished by determining which measurement noise strength values provided the best position estimates. The values are shown in Tables 4.1 through 4.4 and the plots are provided in Appendices F and G. These results provide a basis for comparison of the actual hardware integration and simulation results.

4.2.2.1 69-State Filter Model. As expected, the 69-state model is not perfect. If the 69-state model directly mirrored the real world, then the hardware integration tuning values

would have been relatively close to the those obtained in simulation only. Table 4.1 and Table 4.2 show the differences in dynamic driving and sensor noise statistic values between Category I and II computer simulations. The most glaring differences in the Category I and Category II tuning values are the sensor and dynamic noises applied to the user clock states. The Category I values were very similar to values obtained in past AFIT research (18). The large difference in the clock states' dynamic driving noises and sensor noises could be that the models were developed for P-code receivers. Now that selective availability is a factor, the accuracy is certainly much less. Also, the real world clock is less accurate than had been expected.

The altitude states' dynamic driving noise values show significant differences which require comment. Recall from Chapter I, the Assumptions stated that since no barometric altimeter was available in the laboratory, a barometric altimeter output would be simulated. In actuality, the surveyed altitude of the GPS receiver antenna was fed directly to the INS vertical channel states as the barometric altitude. This simulated input to the INS makes this portion of the hardware integration more unrealistic than the computer simulation. The vertical channel in the hardware integration under this circumstance is extremely accurate and thus the small values of Q. When AFIT moves to a mobile real-time system and determines a source of stabilizing the vertical channel, the Q values for the altitude states will most certainly increase.

The dynamics driving noise values associated with the velocity states actually decreased. This reduction in uncertainty is associated with the stationary platform used in this research. Future AFIT systems will probably need the velocity error states' Q values increased to open up the filter for mobile scenarios. When an integrated navigation system is placed in a mobile environment, coefficients in the system truth model's system dynamics matrix, $\mathbf{F}$, may have been ignored in the reduction to a 13-state filter. This filter order reduction in combination with a moving platform may dictate that more dynamic driving noise be added.

The plots showing the results of using the actual tuning values in the 69-state filter are in Appendix F. The position error plots are shown in Figure 4.3. The filter-predicted one sigma values still bound the mean predicted error and the mean ± one sigma values; however, the filter-predicted one sigma values have grown from a steady-state value of approximately 25 ft to about 100 ft.

Table 4.1 Dynamic Driving Noise Values For 69-State Filter

Element of Q	Category I: 69-State	Category II: 69-State
(1,1)	1×10^{-15}	1×10^{-15}
(2,2)	1×10^{-15}	1×10^{-15}
(3,3)	1×10^{-15}	1×10^{-15}
(4,4)	9×10^{-11}	1×10^{-4}
(5,5)	9×10^{-11}	1×10^{-4}
(6,6)	9×10^{-11}	1×10^{-4}
(7,7)	1×10^{-2}	1×10^{-7}
(8,8)	1×10^{-2}	1×10^{-7}
(9,9)	1×10^{-2}	1×10^{-7}
(10,10)	200	1×10^{-6}
(11,11)	100	1×10^{-6}
(12,12)	1	150000
(13,13)	1×10^{-3}	10000
(14,14) thru (41,41)	1×10^{-11}	1×10^{-12}
(42,42)	1×10^{-5}	1×10^{-9}
(43,43)	1×10^{-5}	1×10^{-9}
(44,44)	1×10^{-5}	1×10^{-9}
(45,45)	1×10^{-5}	5
(46,46)	1×10^{-5}	5
(47,47)	1×10^{-5}	5
(48,48)	1×10^{-5}	5
(49,49)	1×10^{-5}	1×10^{-9}
(50,50)	1×10^{-5}	1×10^{-9}
(51,51)	1×10^{-5}	1×10^{-9}
(52,52)	1×10^{-5}	5
(53,53)	1×10^{-5}	5
(54,54)	1×10^{-5}	5
(55,55)	1×10^{-5}	5
(56,56)	1×10^{-5}	1×10^{-9}
(57,57)	1×10^{-5}	1×10^{-9}
(58,58)	1×10^{-5}	1×10^{-9}
(59,59)	1×10^{-5}	5
(60,60)	1×10^{-5}	5
(61,61)	1×10^{-5}	5
(62,62)	1×10^{-5}	5
(63,63)	1×10^{-5}	1×10^{-9}
(64,64)	1×10^{-5}	1×10^{-9}
(65,65)	1×10^{-5}	1×10^{-9}
(66,66)	1×10^{-5}	5
(67,67)	1×10^{-5}	5
(68,68)	1×10^{-5}	5
(69,69)	1×10^{-5}	5

Table 4.2   Sensor Measurement Noise Values For 69-State Filter

Element of R	Cat I: 69-State	Cat II: 69-State
(1,1)	100	200000
(2,2)	100	200000
(3,3)	100	200000
(4,4)	100	200000

Table 4.3   Dynamic Driving Noise Values For 13-State Filter

Element of Q	Category I: 13-State	Category II: 13-State
(1,1)	5×10^{-14}	1×10^{-15}
(2,2)	5×10^{-14}	1×10^{-15}
(3,3)	5×10^{-14}	1×10^{-15}
(4,4)	9.52×10^{-12}	1×10^{-4}
(5,5)	9.52×10^{-12}	1×10^{-4}
(6,6)	9.52×10^{-12}	1×10^{-4}
(7,7)	0.001	1×10^{-7}
(8,8)	0.001	1×10^{-7}
(9,9)	0.0823	1×10^{-7}
(10,10)	200	1×10^{-6}
(11,11)	166.7	1×10^{-6}
(12,12)	1	150000
(13,13)	1×10^{-3}	10000

4.2.2.2   13-State Filter Model.   The 13-state filter model had results very similar to those of the 69-state model. Tables 4.3 and 4.4 presents the different values for the dynamic driving and the sensor noise strengths.

Tables 4.2 and 4.4 show that the sensor noise variances decreased by a factor of two in going from the 69-state model to the 13-state model. The lower amount of sensor noise in the 13-state model indicates that the EKF is trusting the measurements for the reduced order 13-state model more in comparison to the 69-state model. The filter could also have been tuned by increasing the dynamics driving noise strength values for the 13-state model.

Table 4.4   Sensor Measurement Noise Values For 13-State Filter

Element of R	Cat I: 13-State	Cat II: 13-State
(1,1)	100	100000
(2,2)	100	100000
(3,3)	100	100000
(4,4)	100	100000

Latitude Error (ft)

Longitude Error (ft)
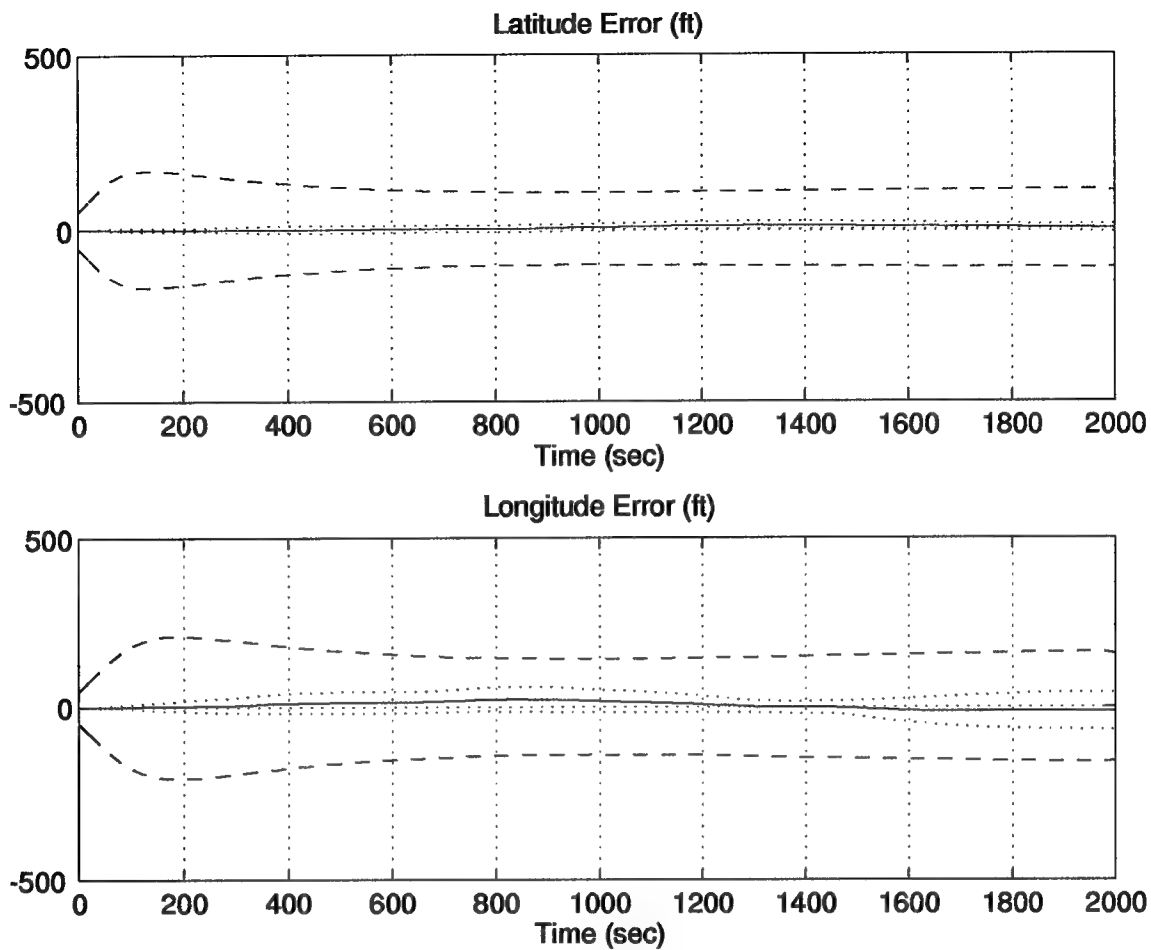
Figure 4.3   Category II: Position Errors for the 69-State Filter Computer Simulation

The position plots for the Category II 13-state filter plots are shown in Figure 4.4. All of these plots can be seen in Appendix G.

4.3   Data Collection

This section describes the position and velocity estimates of the LN-94 and the GPS receivers. A three-hour data set was collected for the INS. The three hours of data collection began after the INS alignment mode had been completed. Several sets of GPS data were taken for both the XR-4PC and XR-5PC over a span of several days. Representative sets are shown in this thesis. The individual subsystem position estimates are important to show the effectiveness of the Kalman filter and its improvement in accuracy over either system alone.
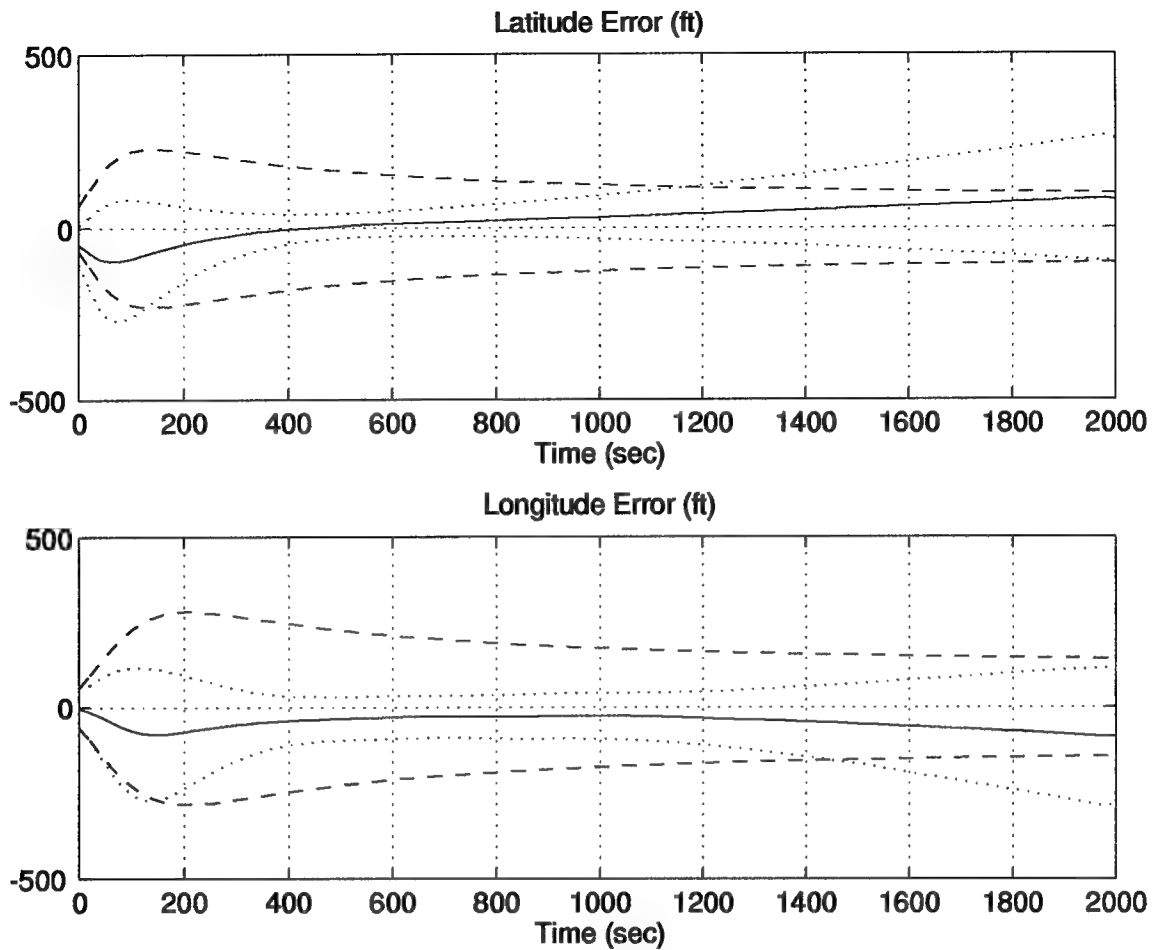
Figure 4.4   Category II: Position Errors for the 13-State Filter Computer Simulation

In order to compare each filter's performance against the others, a temporally averaged mean error was calculated. The temporally averaged mean error for latitude, longitude, and altitude was squared, summed together, and the square root was taken. This left a scalar term which was used to compare each filter's performance. The temporally averaged mean error was calculated for each of the five filters over the ten runs, and is displayed in Table 4.5. The average was taken over all time and the initial transients were included. This method was selected in anticipation of future implementation of a filter in the feedback implementation. The size and duration of the transients would be a factor in determining when the initial reset should take place. In addition, the temporally averaged mean errors for the XR-4PC and XR-5PC are shown.

4.3.1 LN-94 Data. Three hours of data was collected off of the 1553 data bus from the LN-94, after an initial alignment phase had been completed. The LN-94 has three alignment modes: gyrocompass alignment, stored heading alignment, and in-flight alignment. The alignment mode used in this particular research was the gyrocompass alignment mode. The gyrocompass alignment mode allows the INS to achieve the highest initial alignment accuracy of the three modes. Data was not taken until the gyrocompass align mode was complete. After one hour the LN-94 drifted approximately 0.84 nm. This is a relatively high drift, since the .8 nm/hr specification rate is the worst case for an operational system; however, this particular LN-94 has undergone no calibration or maintenance in over four years. Thus a relatively high drift rate is expected.

The complete three-hour data plot is shown in this section to illustrate the low frequency error characteristics of inertial navigation systems. The sinusoid riding the sloping curve is actually the Schuler oscillation. The Schuler Frequency has a period of approximately 84 minutes. The LN-94 output is also modulated by the Earth and Foucault rates. Since these rates have much larger periods, a longer data set would be necessary to see their effects.

Figure 4.5 clearly illustrates the need for GPS updates to the INS, especially in missions of longer durations. The average temporal error for the INS over the 1000 second interval was 1415.2 feet. All of the INS plots are shown in Appendix H.

4.3.2 XR-4PC Data. The XR-4PC is a two-channel receiver and can only receive C/A code; therefore, selective availability degrades the navigation performance. Five sets of data were taken over a three day period for the XR-4PC; Figure 4.6 shows the mean ± one sigma for position.

All five XR-4PC data sets used seven to eight satellites in computing a position estimate. The positions were found by using the batch least squares method described in Chapter II. The temporally averaged statistic error for the five sets of XR-4PC data was 252.42 feet.

4.3.3 XR-5PC Data. The XR-5PC is a six-channel receiver and is also subject to selective availability. Five sets of data were also taken for the XR-5PC; the data was taken at different hours over a four day period. The mean ± one sigma deviations of position are shown in Figure 4.7.

The six-channel XR-5PC receiver is much quicker than the two-channel XR-4PC in acquiring satellites for positioning. Also the position estimates are much smoother than those produced by the XR-4PC, yet the same method is used to calculate a GPS-only solution, the batch least squares.

Latitude Error

Longitude Error
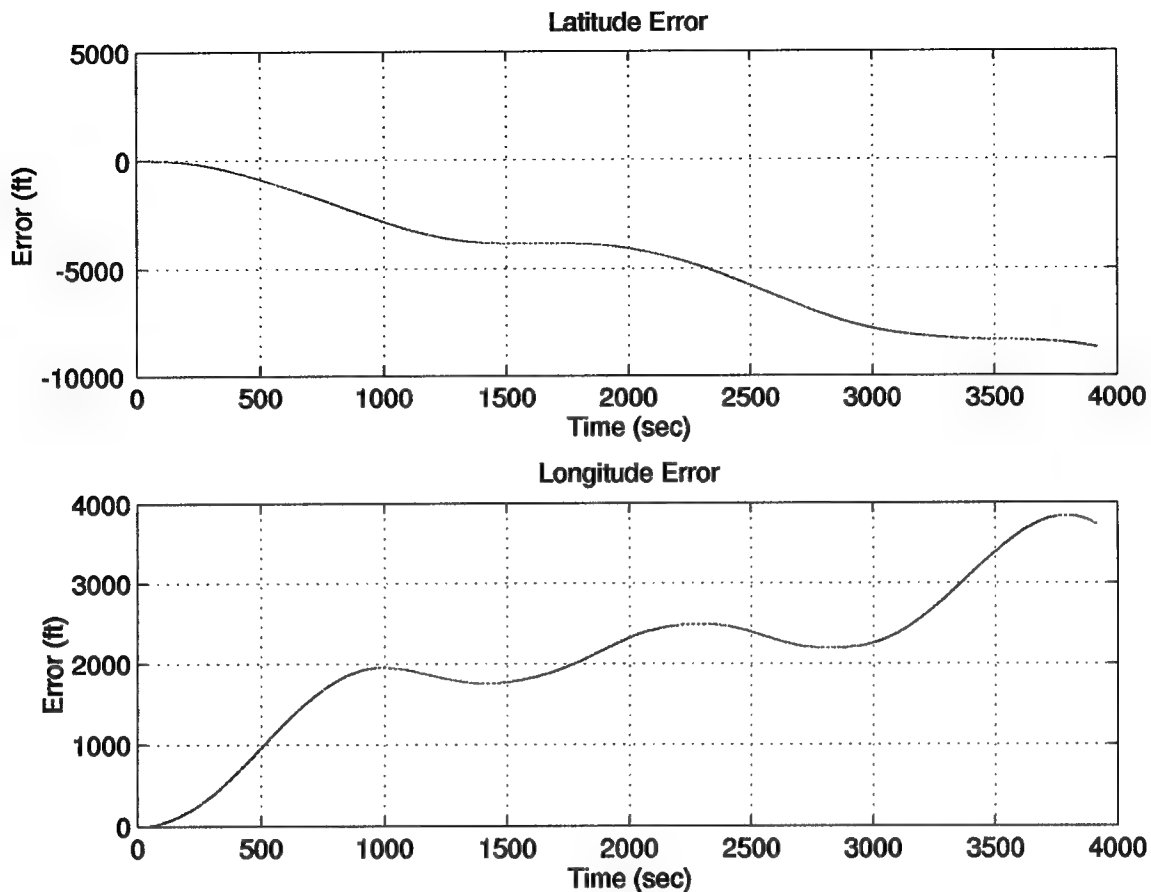
Figure 4.5   LN-94 Position Errors

The accuracy of the XR-5PC was poorer than the XR-4PC; the temporally averaged statistic error was 318.34 feet. The XR-5PC locked up only five to seven satellites in the five data sets collected. The algorithms used within the two receivers are most certainly different. It seems the two-channel XR-4PC cycles through all available satellites, while the manufacturer, NAVSTAR, chose to allow the six-channel, XR-5PC, to dedicate channels to satellites.

4.4   GPS/INS Integration

This section divides the results into three categories: a comparison of the actual results to the various simulations, a comparison of the five different filters, and the performance of the EKF in determining the position versus the individual system's estimates.
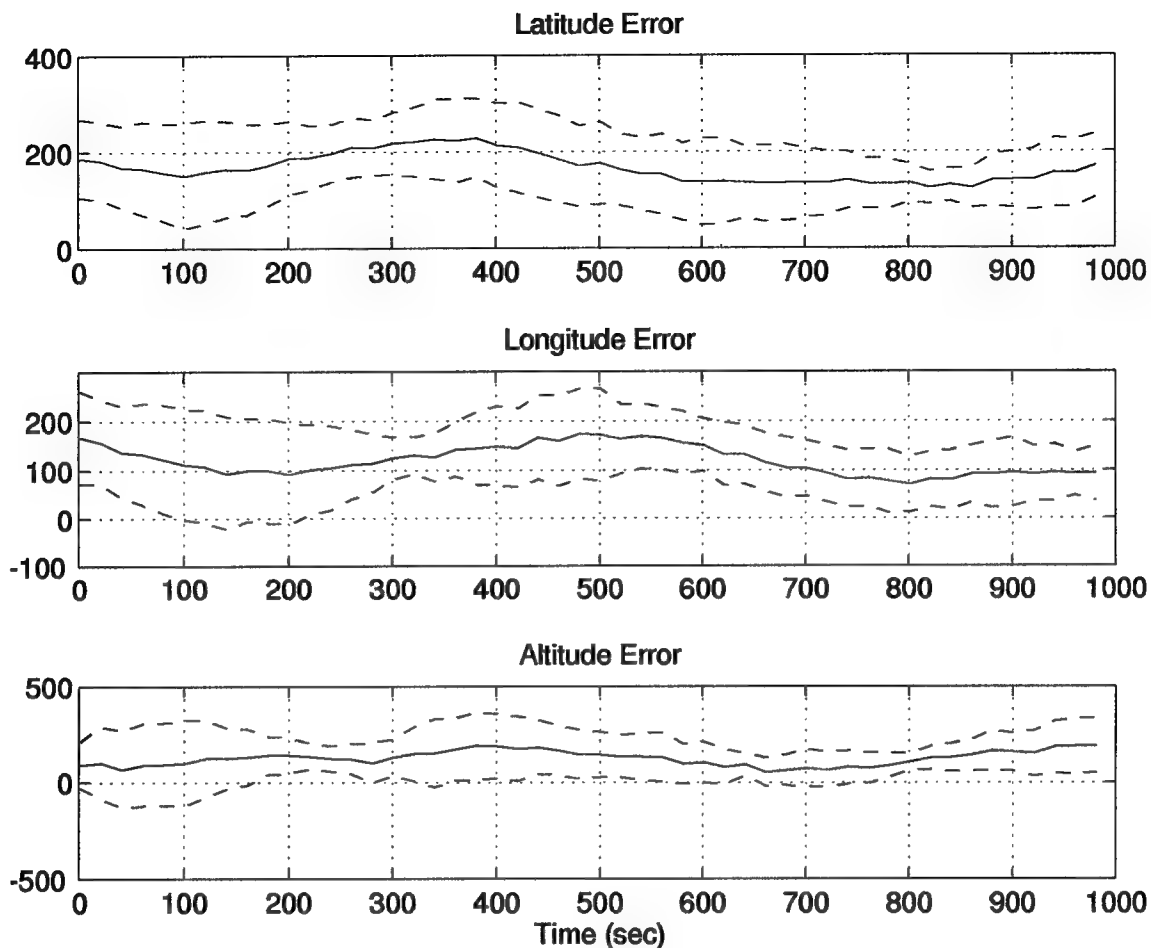
Figure 4.6   XR-4PC Position Errors

4.4.1   Comparison of Actual Results to the Simulations.   The following section shows the results of the hardware integration. A total of ten feedforward tightly-coupled GPS/INS integration runs were accomplished using the five data sets of XR-4PC data and the other five files collected from the XR-5PC. The mean and one sigma deviations were computed for these ten data sets and the results are plotted in Figures 4.8 and 4.9 for both the 69-state and 13-state filters. The tuning values were presented in Tables 4.1 through 4.4; all tuning parameters are Category II. When the Category I tuning values were used in the hardware integration, the EKF gave divergent results.

The 69-state filter Category I computer simulation produced a mean of zero with a mean ±10 feet in the latitude and ±40 feet in the longitude. The Category II simulation was similar to Category I except the filter predicted ±sigma went from 25 feet to nearly 100 feet. The hardware integration results showed that, after the 100 second point, the mean is no longer zero as with the
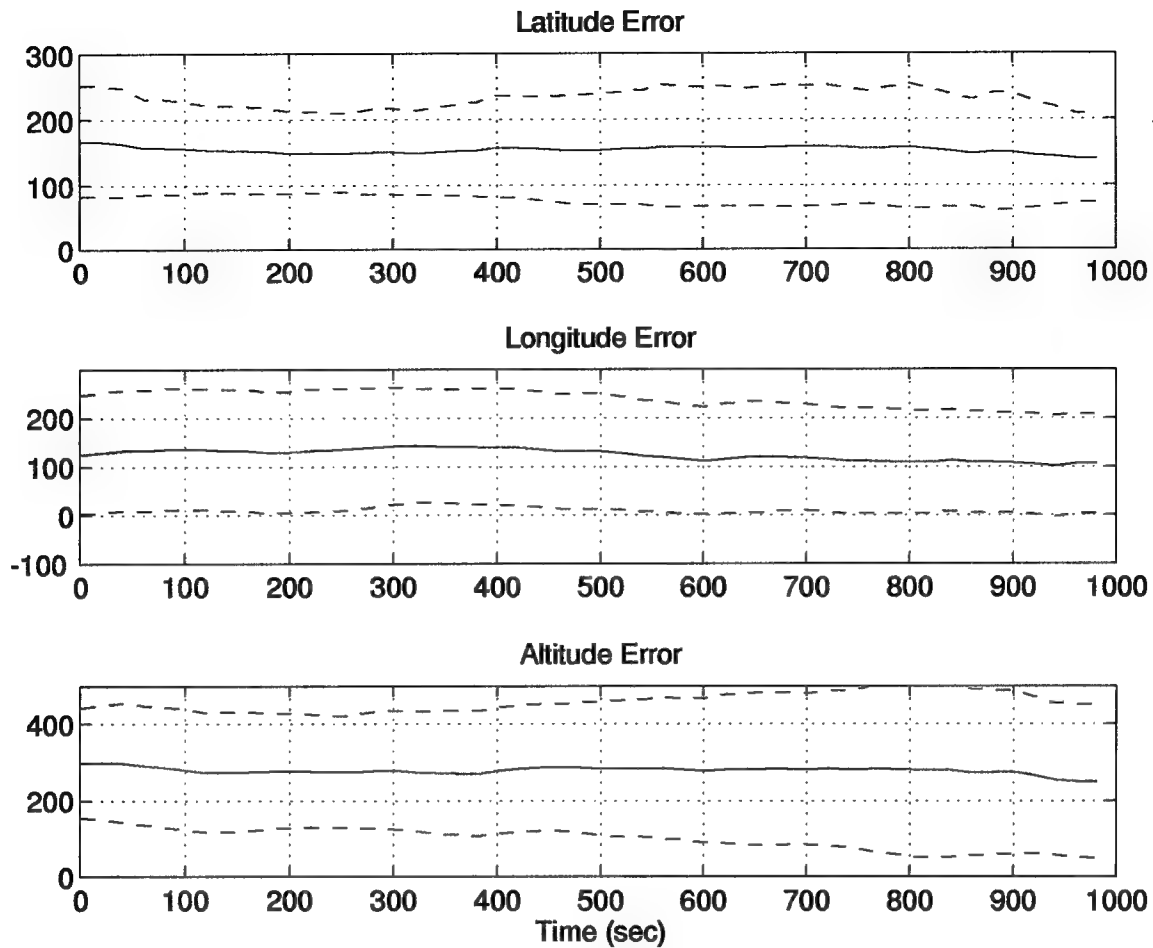
Figure 4.7 XR-5PC Position Errors

simulations. This drifting from zero error indicates that the 69-state filter is no longer modeling the actual errors correctly. The 69-state errors are possibly a result of modeling the nonlinear error characteristics of the actual hardware with a linear model. The mean ±sigma is 150 feet in the latitude and 100 feet in the longitude. The longitude mean ±sigma envelope remains very constant but the latitude seems to be growing. In any event, both computer simulation results are much more optimistic than the hardware results.

The Category II simulation results for the 13-state filter are much closer to the real data results. The 13-state filter results are very nearly the same as those of the 69-state filter. The major difference is the transient in the altitude error. In fact, the latitude errors for the 13-state filter are closer to zero mean and have a smaller mean ±sigma envelope than the 69-state filter.
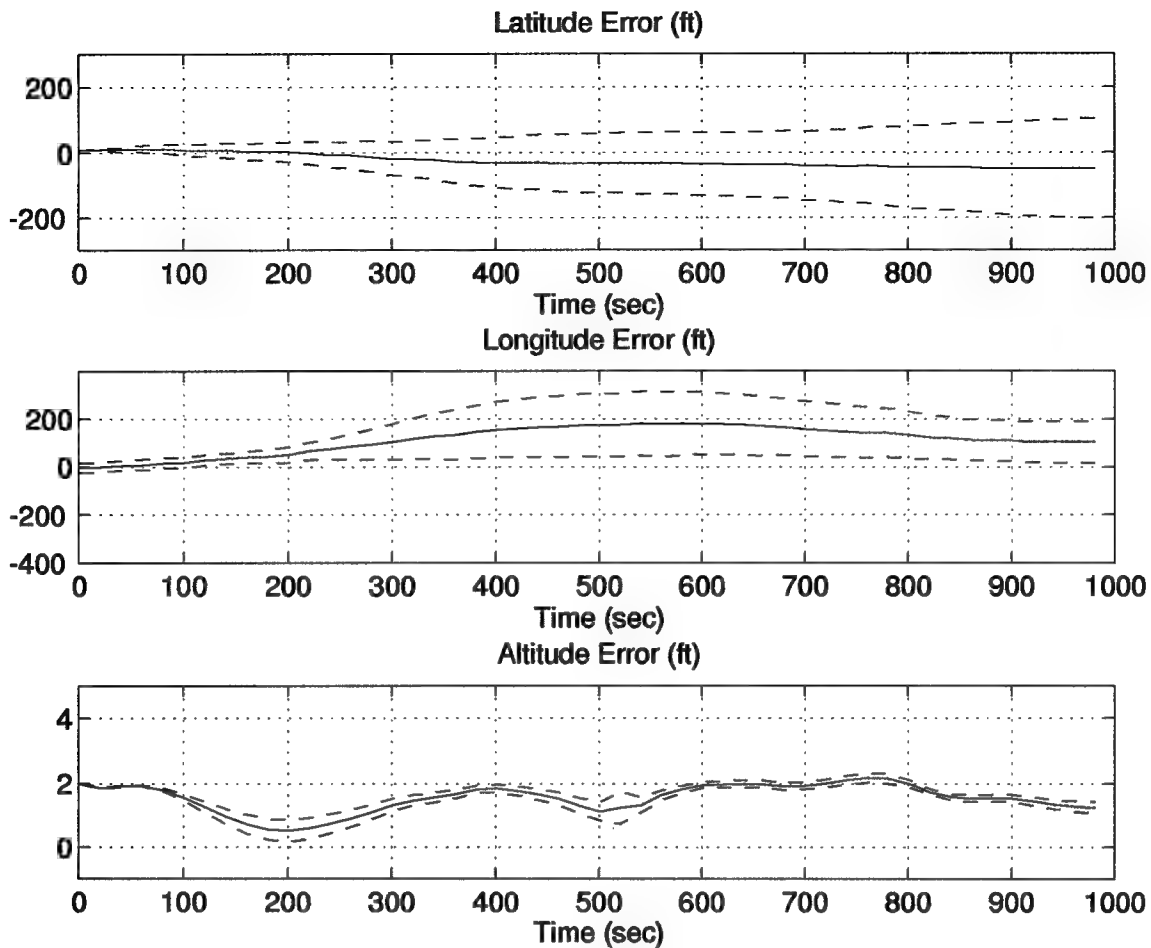
Figure 4.8   EKF Position Errors Using the 69-State Filter

This difference may indicate a tuning problem that is yet to be resolved. The longitude plots are nearly the same.

4.4.2   Comparison Of the 69-State, 61-State, 49-State, 41-State, and 13-State Filter Models.

One objective of this thesis was to determine the accuracy of various GPS models and collect data for analysis. Issues to be resolved were (1) which models would be best for use in the system truth model and (2) how much degradation occurred in reducing the model for use in the filter.

The analysis was conducted by using the five sets of data collected from the XR-4PC and the first 1000 seconds of the LN-94 data. There were five different filters used. For purposes of comparison, the 39-state INS error model was used in four of the filters to reduce errors. The 11-state INS model was used in the 13-state filter because this will most likely be the filter employed

4-16
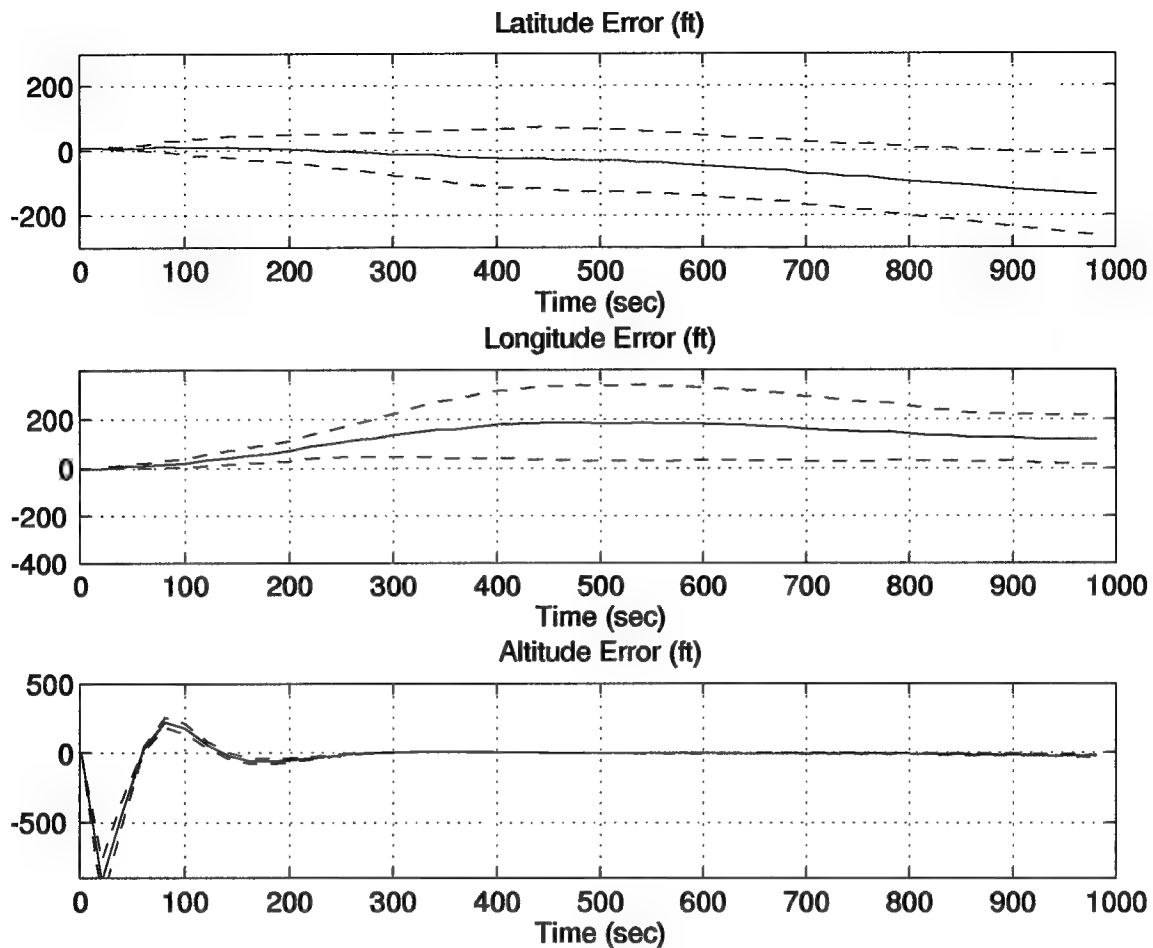
Figure 4.9   EKF Position Errors Using the 13-State Filter

in a real-time application. The 69-state model contains the 30-state GPS model, which is the one used by AFIT in past research (18). The 61-state model contains a reduced order version of the 30-state model. This 61-state model is reduced by eight states by combining the three line-of-sight errors for each satellite into one error. This reasoning is justified since the pseudorange is taken in one line-of-sight vector and not broken into three. The third GPS filter is the 49-state model. The 10-state GPS model was taken from the MSOFE example problem (3) and combined with the 39-state INS model (3). The 41-state model has only two GPS states: the user clock bias and drift. The final filter is the 13-state filter, which also only contains two GPS states (18). Table 4.5 presents a comparison of the five filters' performance.

Surprisingly, the 69-state model did not perform as well as the 61-state and 49-state models. Although the 61-state model had the lowest error over the five runs, it would seem that the 49-

Table 4.5   Filter Comparison

Filter	Temporally Averaged Mean Error (ft)
69-state	118.17
61-state	112.94
49-state	117.27
41-state	137.54
13-state	140.37
XR-4PC (only)	252.42
XR-5PC (only)	318.34

state model may be the best choice for the system truth model. The 49-state model carries less of a computational burden due to its reduced number of states, but still maintains a comparable accuracy with the 61-state model.

Another interesting result was that the temporally averaged mean error statistic between the 41-state and 13-state models was only three feet. The only difference between these two filters was the INS models (39-state versus 11-state). This would indicate that the INS model used in the 13-state filter is nearly as accurate under the stationary short-term circumstance as the 39-state model.

As discussed earlier, tuning is critical to performance. In order to accomplish as fair an evaluation as possible, the 69-, 61-, 49-, and 41-state models all used the same dynamic driving noise values for the INS error states and sensor measurement noise. These values are shown in Table 4.1 and 4.2. The GPS dynamic driving noise values are given in Table 4.6 for future reference. The elements of Q given in Table 4.6 are shown in detail in Appendix A. Different tuning values will obviously change the temporally averaged mean errors shown in Table 4.5.

4.4.3   EKF Results Versus Individual Systems.   The performance of the EKF in comparison to the individual INS and GPS systems definitely proves its worth. Table 4.5 shows the temporally averaged mean errors of the individual systems and each of the five filters used in the EKF. After 1000 seconds the bare INS has drifted nearly 1000 feet in both the latitudinal and longitudinal directions. The XR-4PC and XR-5PC had temporally averaged mean errors of 252.42 feet and 318.34 feet respectively. The EKF greatly reduced the error in the navigation position estimates. The 13-state filter had the worst results of the five filters (temporally averaged mean

Table 4.6   Filter GPS States Tuning Values

Element of Q	69-State	61-State	49-State
(42,42)	1×10^{-9}	1×10^{-9}	12
(43,43)	1×10^{-9}	1×10^{-9}	1×10^{-9}
(44,44)	1×10^{-9}	1×10^{-9}	12
(45,45)	5	5	1×10^{-10}
(46,46)	5	15	12
(47,47)	5	$1x10^{-9}$	1×10^{-10}
(48,48)	5	1×10^{-9}	12
(49,49)	1×10^{-9}	1×10^{-9}	1×10^{-10}
(50,50)	1×10^{-9}	15	N/A
(51,51)	1×10^{-9}	5	N/A
(52,52)	5	1×10^{-9}	N/A
(53,53)	5	1×10^{-9}	N/A
(54,54)	5	1×10^{-9}	N/A
(55,55)	5	5	N/A
(56,56)	1×10^{-9}	15	N/A
(57,57)	1×10^{-9}	1×10^{-9}	N/A
(58,58)	1×10^{-9}	1×10^{-9}	N/A
(59,59)	5	1×10^{-9}	N/A
(60,60)	5	5	N/A
(61,61)	5	15	N/A
(62,62)	5	N/A	N/A
(63,63)	1×10^{-9}	N/A	N/A
(64,64)	1×10^{-9}	N/A	N/A
(65,65)	$1x10^{-9}$	N/A	N/A
(66,66)	5	N/A	N/A
(67,67)	5	N/A	N/A
(68,68)	5	N/A	N/A
(69,69)	5	N/A	N/A

error of 140.37ft) and was still much better than the GPS or INS alone. In order to illustrate the smoothing effect that the EKF has on the position estimates, Figures 4.10 and 4.11 are included.

After an initial transient, the residuals for the hardware integration, (both 69-state filter and 13-state filter), nearly all fell within a ±400 foot envelope after steady state had been reached. Figures 4.12 and 4.13 present seven sets of residuals plotted during one 1000 second run. Figure 4.12 shows the residuals and covariance of the residuals, while 4.13 is just rescaled to show the residuals. The statistics showing the mean and covariance of the residuals for all ten runs could not be easily plotted, because of the shifting of satellites during runs. Also, there were over twenty different satellites used in the data sets. Some were only used for portions of a data set, while data from others were used in some extent in up to five runs. Luckily, the satellites used in Figures 4.12
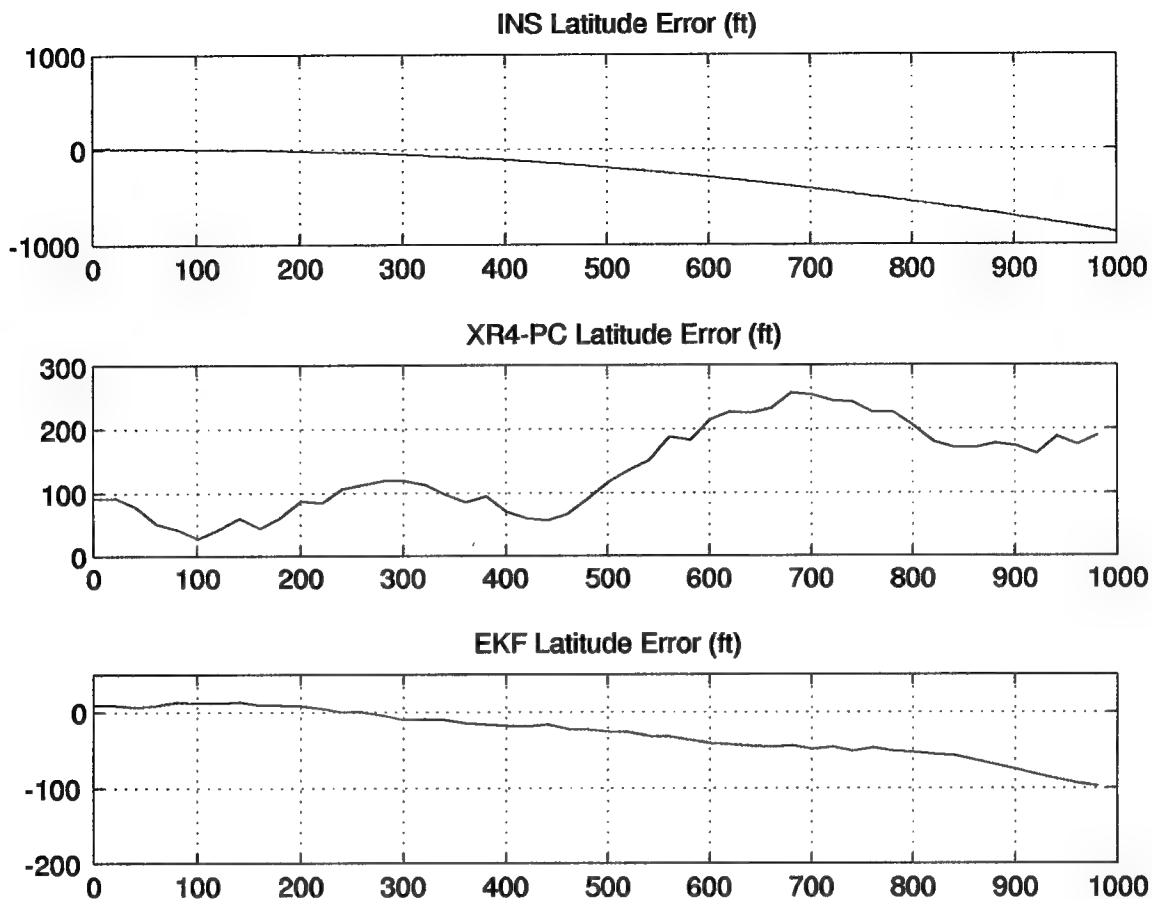
Figure 4.10   INS, GPS, and EKF Latitude Errors Using the 13-State Filter

and 4.13 did not change during the 1000 second interval. The large covariance of residuals, shown in Figure 4.12, was required because of the satellite clock bias shifts and changing satellites. The significantly larger values of sensor measurement noise strengths, shown in Tables 4.2 and 4.4, were required to make the filter more robust. When sensor measurement noises were decreased below those given in Tables 4.2 and 4.4, only data sets with no user clock bias shifts or satellite changes could be used in the EKF.

An observation was made during data collection that at random intervals the GPS receiver clocks applied a bias to the pseudoranges. The bias was consistent among all satellites at the same instant. The EKF could handle nearly all of these bias shifts. The point in time where the bias occurred could not even be seen in the position outputs. The only place the jumps were evident was in the residuals. A large spike would occur and then the EKF residuals would return to steady-state values within three measurement updates. Plots and research data are given in detail in Appendix

INS Longitude Error (ft)

XR4-PC Longitude Error (ft)

EKF Longitude Error (ft)
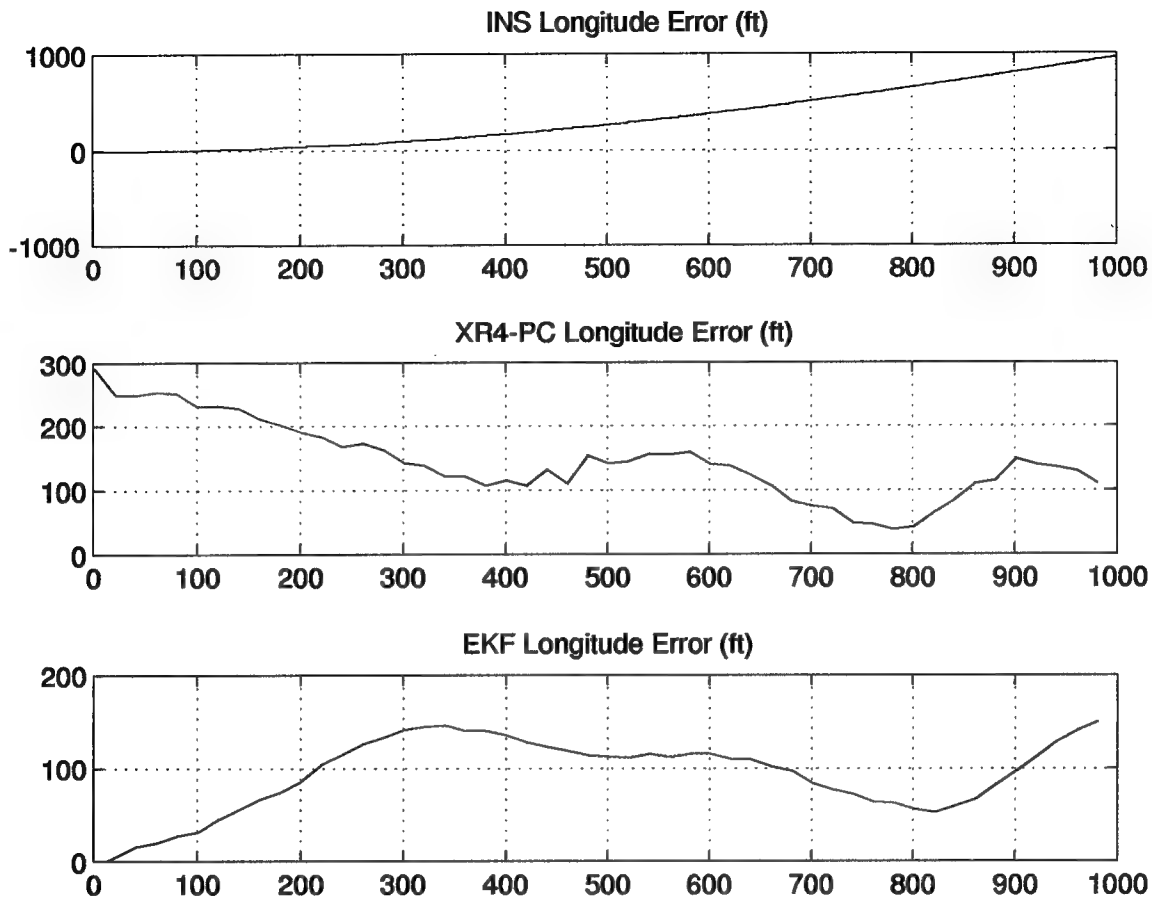
Figure 4.11   INS, GPS, and EKF Longitude Errors Using the 13-State Filter

I. The clock shifts were more noticeable in the XR-4PC than in the XR-5PC. Clock differences may be one reason that the XR-5PC outputs seem smoother than those of the XR-4PC.

The dynamic driving noise values associated with the user clock bias and drift states had the most impact on the residuals. The sensor measurement noise variance also had an impact on the residuals. Tuning of the dynamic driving noise for the clock bias and sensor measurement noises changed the spread of the residuals, and the dynamic driving noise for the clock drift determined how quickly the residuals reached steady-state.

4.4.4   Effects of Measurement Update Rate On Position Estimates.   AFIT's future real-time EKF will be much more sensitive to update rates than for this research. The computational capacity will be limited and the timing intervals more critical. The mobile system will employ a 486/33MHz laptop computer, while the hardware integration for this research has been conducted
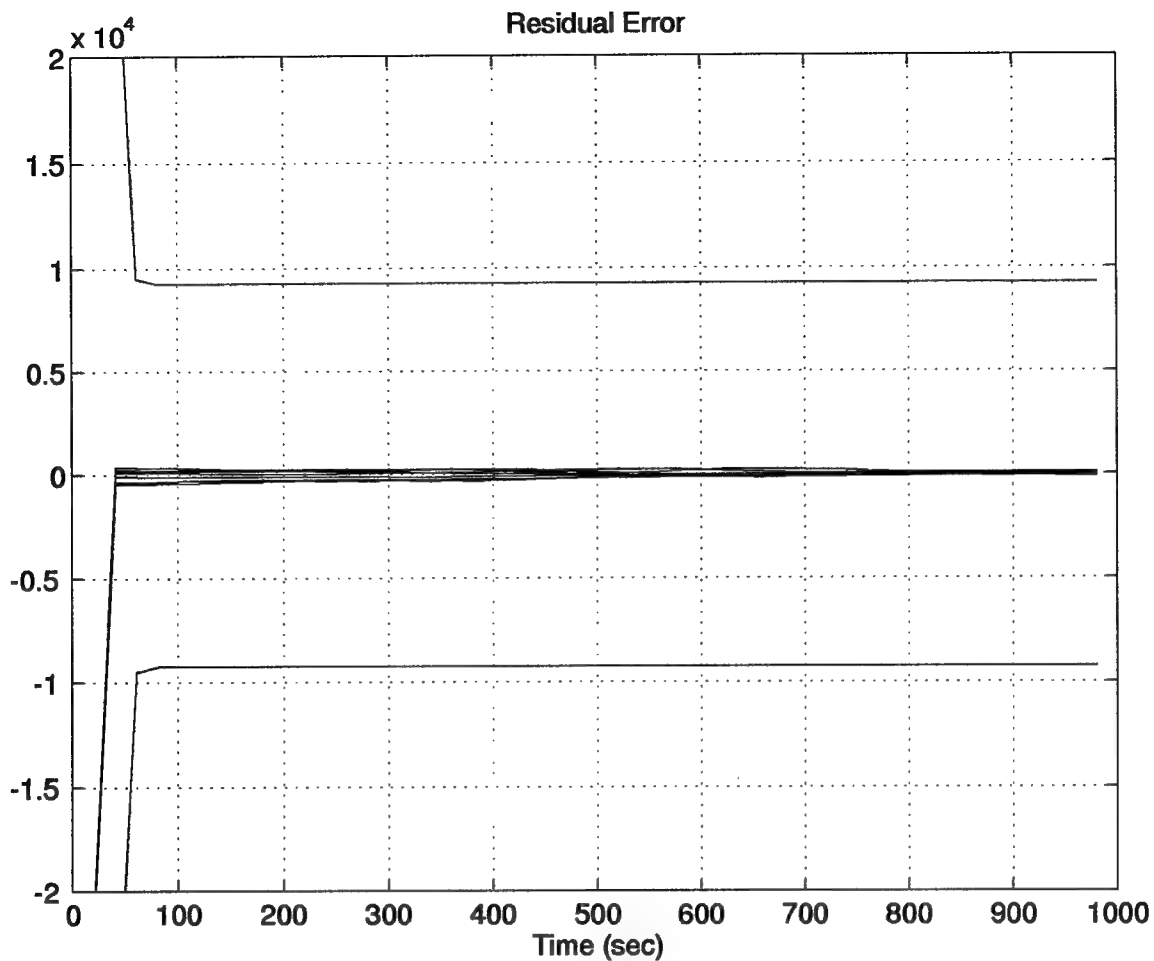
Figure 4.12 Seven Residuals And Covariances of Residuals of the 69-State Filter During A 1000 Second Run

on a Sun Sparc20 workstation. The measurement update rate for all data taken thus far has been once every 20 seconds. A study was accomplished to determine the effects that the update rate has on the position estimates. By taking measurements at a higher frequency, the Kalman filter has less time to propagate before the next update. While a greater frequency of updates should improve accuracy, it most certainly will increase computation time.

In order to determine how much accuracy could be gained, the measurement update periods were halved from 20 seconds to 10 seconds. The 13-state filter was used for this study and a data set was collected from the XR-4PC. As with all of the previous integrations in this research, the filter was tuned for a 1000 second interval. The filter was tuned for 1000 seconds because this was the length of the GPS/INS integration run used in the research. Interestingly, there was a significant

4-22

Residual Error
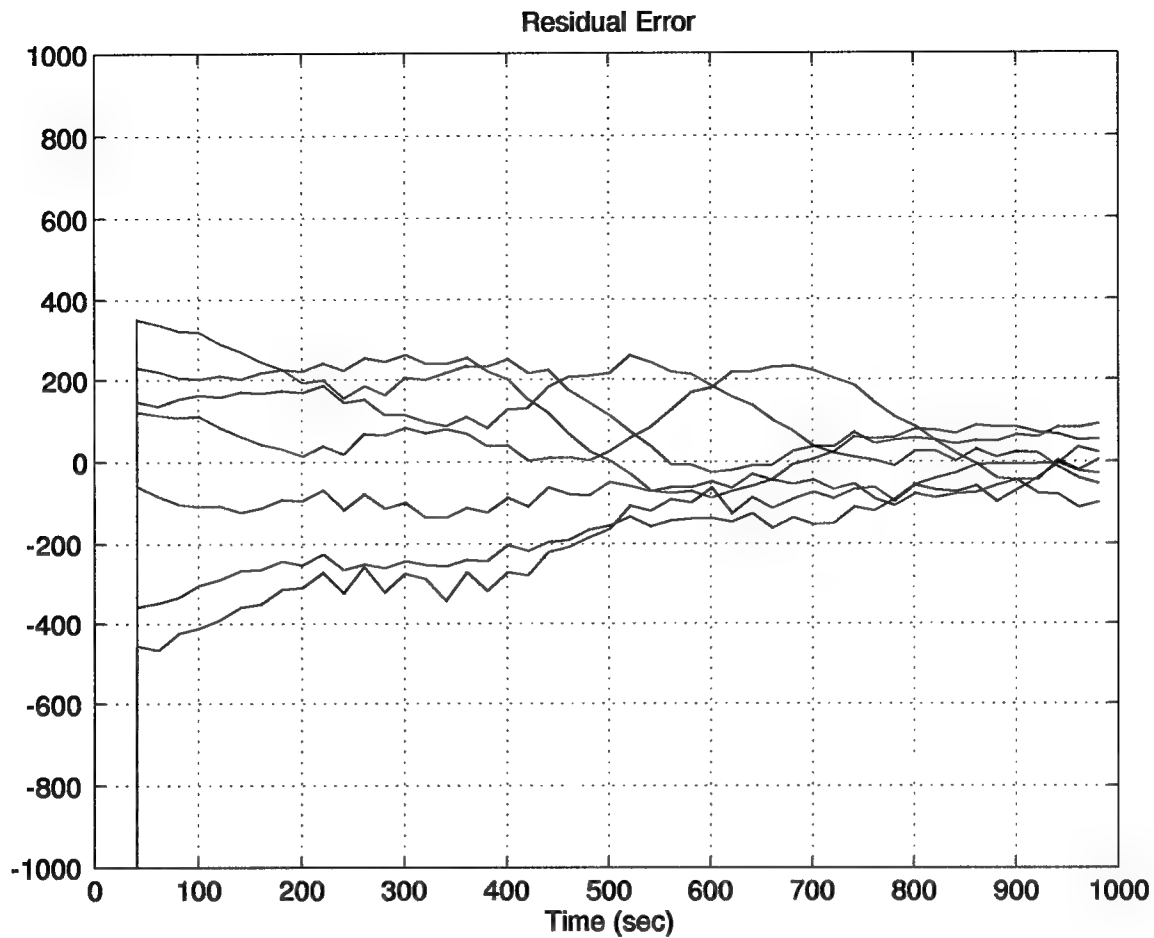
Figure 4.13   Seven Residuals of the 69-State Filter During A 1000 Second Run

difference in tuning values between the 20-second and 10-second measurement update GPS/INS integrations. The system using 20-second updates had the same tuning values as were previously given in Tables 4.3 and 4.4. The system employing the 10-second measurement updates had the same dynamic driving noise strength values, Q; however, the sensor noise, R, changed drastically. The optimum filter performance was found by increasing R from 100,000 to 500,000. The dynamic driving noise strength, Q, is the actual value tuned. The discrete value of Q, Q_d, is calculated in the MATSOFE algorithms. The sensor and dynmamic driving noise strengths should not change as the measurement update periods are decreased or increased. Figures 4.13 and 4.14, show the residuals for the seven different satellites during the run and the covariance of the residuals. The required increase in sensor noise strength for the filter employing the 10 second measurement update period may be due to the satellite change at 250 seconds and the clock bias shift at 790 seconds. Table

4-23

Residual Error
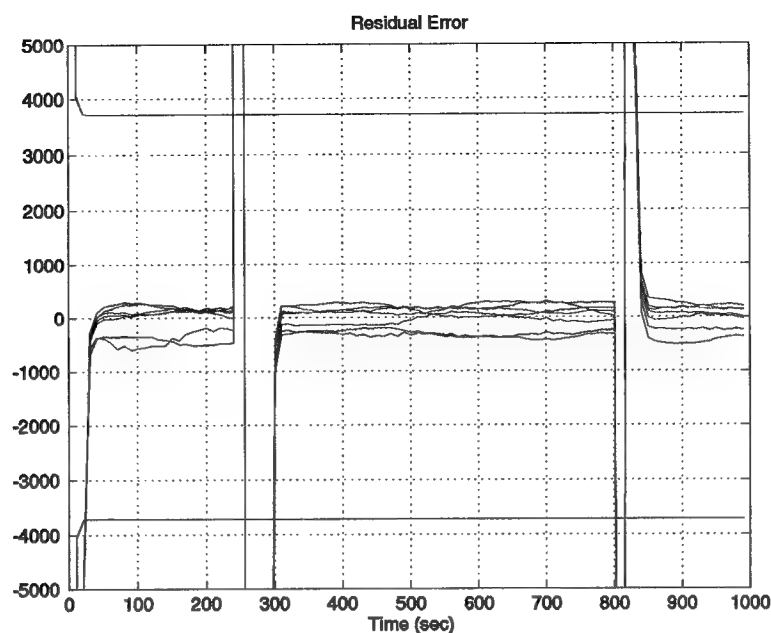
Figure 4.14   Residuals and Covariances of Residuals For The 10 Second Measurement Update Period

4.7 presents the results. The results permit this researcher to believe that the increase in accuracy is not worth the greater computational burden placed upon the filter. However, this is one decision the developer of a GPS/INS system must make and is contingent upon the situation.

Table 4.7   Comparison of Measurement Update Rates

Parameter	20 sec	10 sec	GPS-only solution
Latitude	41.80	21.0818	162.38
Longitude	91.31	81.27	161.98
Altitude	42.10	51.66 7	128.49
Temporally Averaged Error	108.89	98.58	262.90

4.4.5   Preliminary Results For Use In Implementing a Feedback GPS/INS Integration.
As previously stated, this research is the first in a series of topics aimed at developing a mobile real-time integrated GPS-INS navigation system at AFIT. One topic of particular interest is the advantages and disadvantages of feedback versus feedforward approaches. The theory for these EKF implementations is given in Chapter II. AFIT research has primarily concentrated on the feedforward approach, even with its known drawbacks. The main reason for this is that the feedforward allows the data to be collected from individual subsystems and then post-processed as is
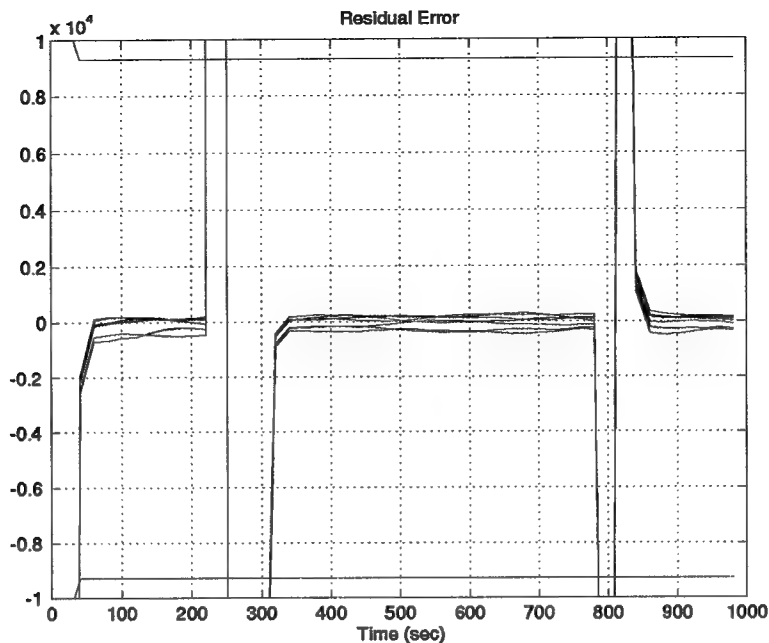
Figure 4.15   Residuals and Covariances of Residuals For The 20 Second Measurement Update Period

done with CIGTF's Navigation Reference System or in this thesis. The feedforward approach is also known as an open-loop system; thus if the EKF has a failure of some sort the INS outputs are not affected. In an indirect feedback system or closed-loop system, the INS is reset periodically by the EKF. Because of the drift characteristics of the INS, resetting the INS periodically should greatly improve results, as long as the resets are not grossly inaccurate.

The filter can be tuned for different intervals. For intervals of short duration, the filters can be tuned to rely heavily on the model because it still closely approximates the INS errors. As the interval grows and the INS drifts further away from the actual point, the EKF can be tuned to maintain a certain level of accuracy. The EKF may be less accurate over the beginning of the interval but throughout the whole length of the run the position estimates are better. The sensor noise is decreased to rely more and more on the measurements rather than the model. However, eventually performance will be degraded to nothing more than the GPS-only solution. In the scalar case, the Q_d/R ratio is important for establishing gain. When tuning the filter, the reduction in sensor noise is not saying that the sensor measurements have become more accurate; the dynamic driving noise could have been increased.

Latitude Error (ft)

Longitude Error (ft)
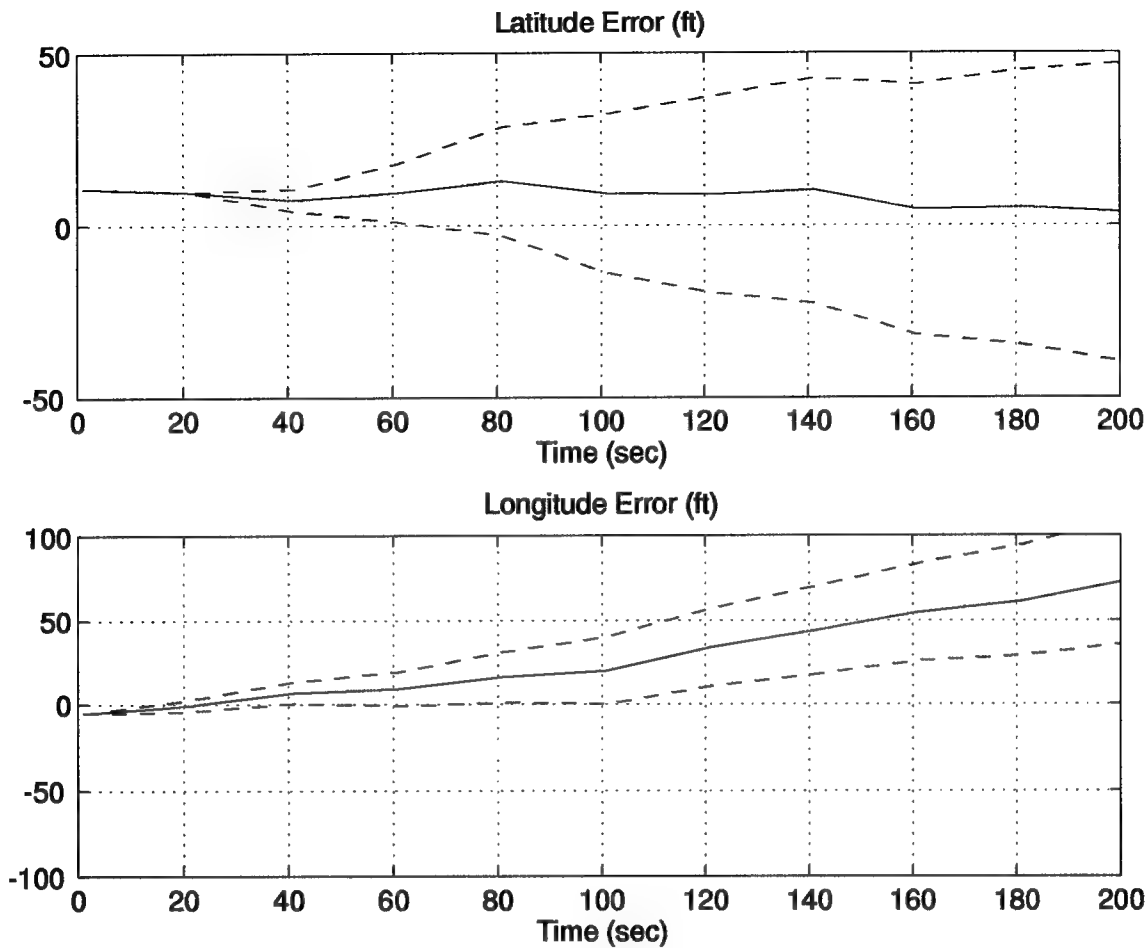
Figure 4.16   EKF Position Errors Using the 13-State Filter (First 200 seconds of Fig. 4.9)

The first 200 seconds of Figure 4.9 are shown in the following plots. The filter of Figure 4.16 was tuned for the specific interval of 1000 seconds to produce the best possible results. In Figure 4.16, the EKF update of the longitude error begins to have less of an impact at the 100 second point, and the plot seems to begin divergence. These plots would indicate resetting the latitude and longitude positions of the INS at least every 100 seconds, if not more often. The accuracy attained by moving to a feedback implementation is much greater. For the ten runs the mean error ± one standard deviation after the first 100 seconds of the run is no more than 50 feet in both latitude and longitude.

An interesting result was found when trying to determine where feedback surpassed feedforward results. A three-hour run was conducted with LN-94 and XR-4PC data. The XR-4PC data was actually a one-hour file duplicated three times for the EKF. Evidence of this tripling of the

data file can be seen in Figure 4.17 by the short spikes in the latitude and longitude plots. The LN-94 data is shown in Appendix H. The 1000-second duration tuning values were used for both the 69- and 13-state filters. The results of these runs are shown in Figures 4.17 and 4.18.
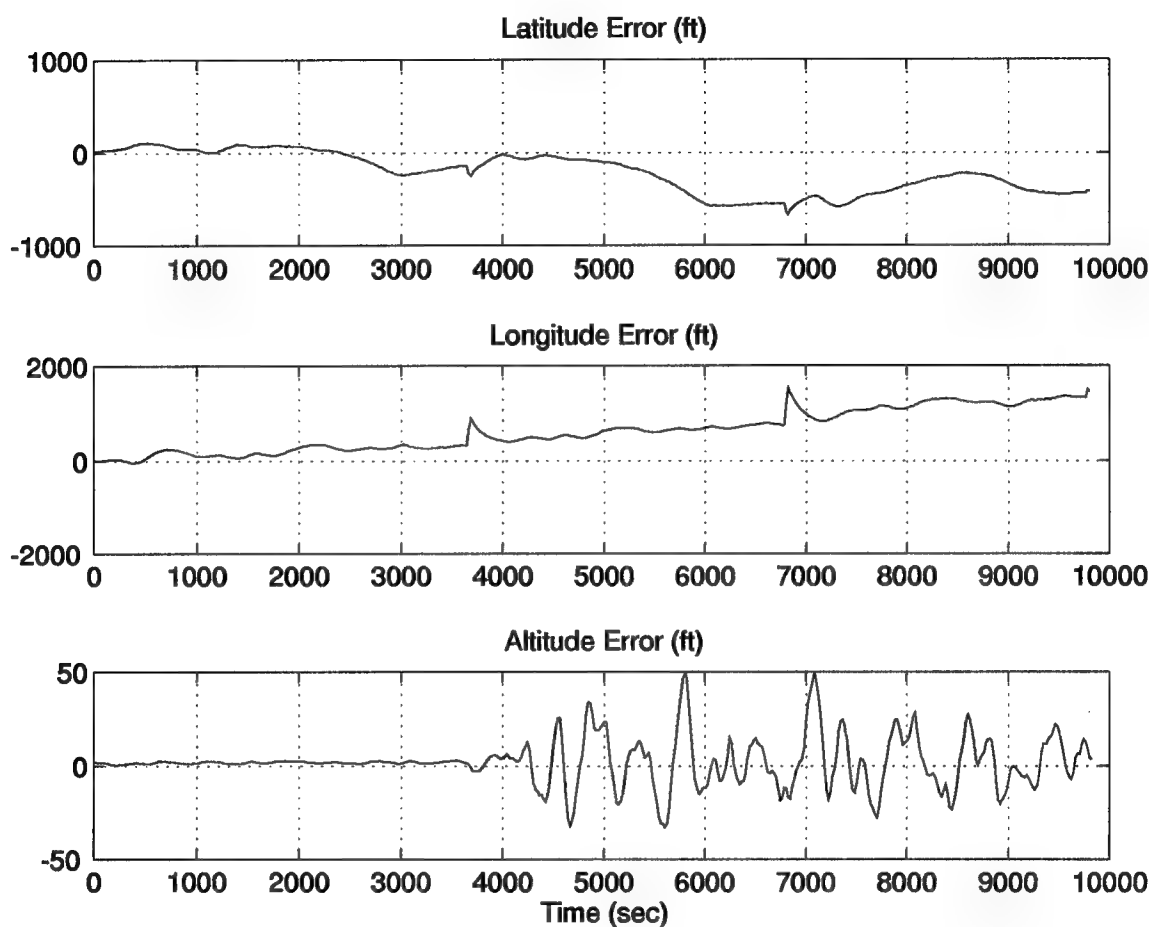
Figure 4.17   Three Hour Run EKF Position Errors Using the 69-State Filter

Oddly enough, the 13-state filter when tuned for 1000 second runs outperformed the 69-state filter. The temporally averaged mean error for the 13-state filter was 378.54 feet, while the 69-state filter had a temporally averaged mean error of 677.07 feet. The assumption was made that, if tuned for the three-hour run, the 69-state model would outperform the 13-state filter. This was not to be the case. The 69-state filter sensor measurement noise variance was lowered to 100 and the temporally averaged mean error was only decreased to 424.65 feet. This situation seems to be one where the INS errors in the long term are overmodeled with the 39-state INS model. In the 13-state model there is more uncertainty and therefore, more dynamic driving noise added to
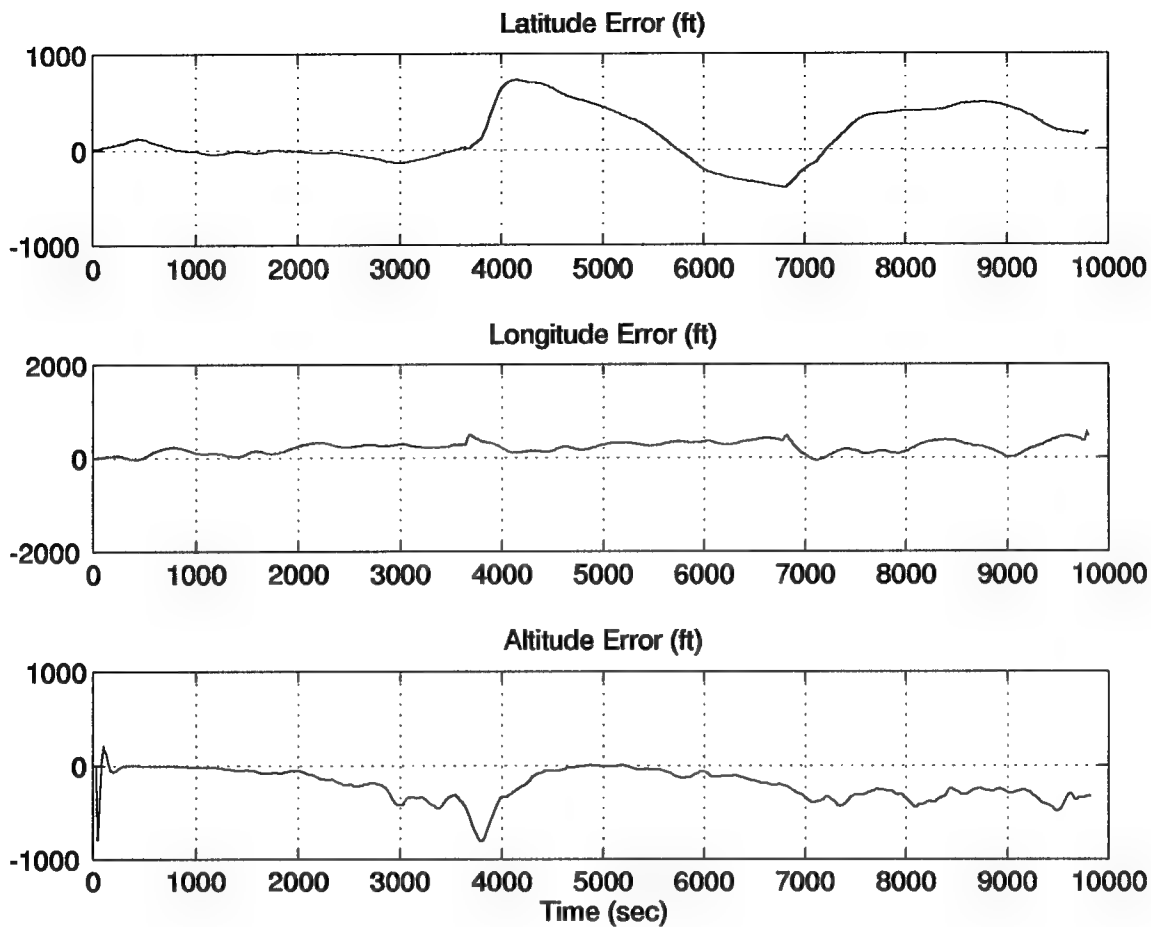
Figure 4.18   Three Hour Run EKF Position Errors Using the 13-State Filter

the filter. This additional noise allows the filter flexibility in following unmodeled errors. The possibility exists that over longer periods of time the 39-state filter misrepresents some of the INS errors and cannot compensate for the actual errors as well as the 13-state filter. The temporally averaged mean error for the GPS-only solution is 267.78 feet; therefore the 69-state filter could not even be tuned to outperform the GPS. As the sensor noise variance neared perfect measurement values the estimates should mirror those of the GPS receiver.

Table 4.8 shows the results for the integration of an LN-94 and an XR-4PC tuned for a 1000 second run. Table 4.8 and Figures 4.17 and 4.18 illustrate where the 69- and 13-state models break down. The longitude slowly slopes off and after one hour the 13-state provides better position estimates than the 69-state filter. The residuals and covariances of the residuals for the three hour run are shown in Figures 4.20 and 4.21 for the 69-state and 13-state filters respectively. The

4-28
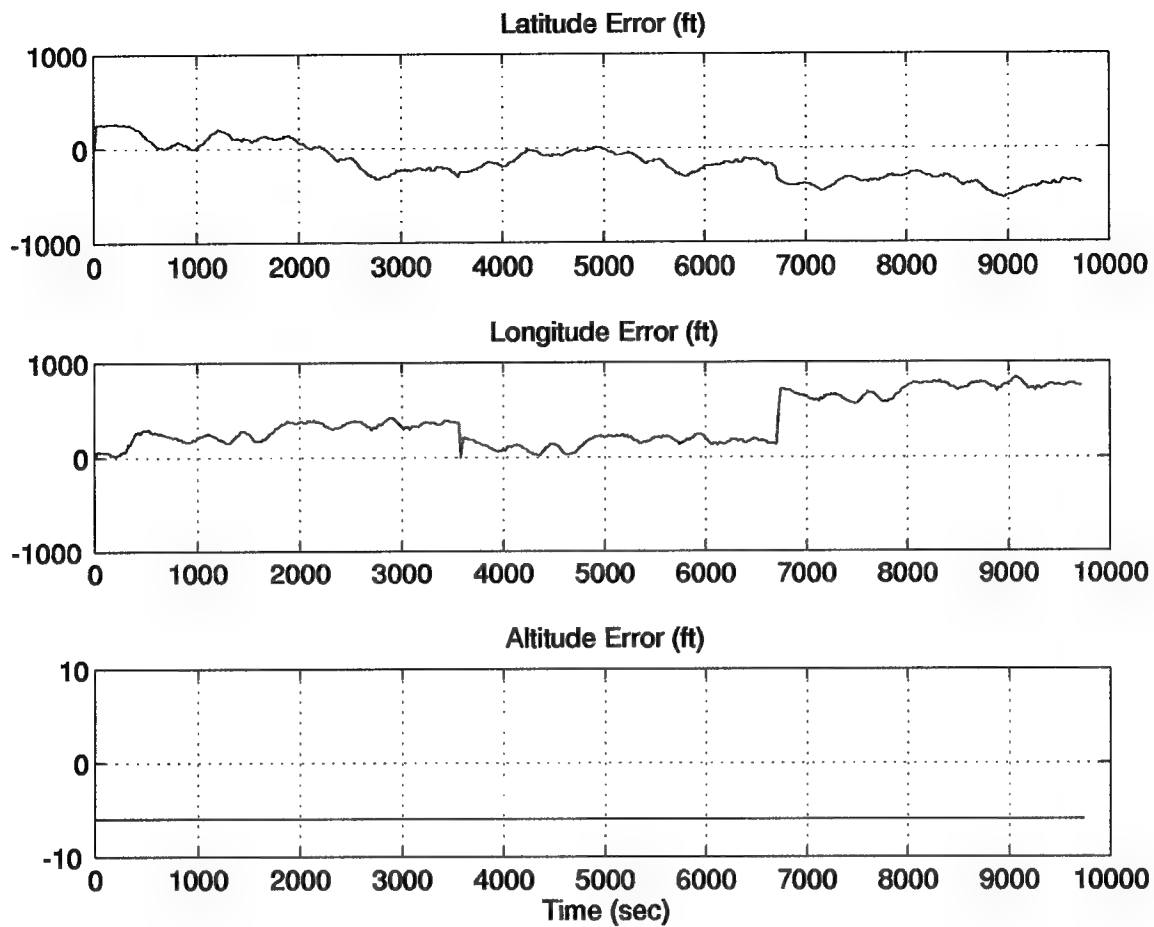
Figure 4.19   EKF Position Errors Using the 69-State Filter (Tuned for Three Hour Run)

sharp spikes at the 3600 second and 6800 second points are from the splicing of the data sets; the remaining spikes are from receiver clock bias shifts.

Table 4.8   Temporal Errors At Specific Times During a Three Hour Run

Filter	1000 sec	1 hour	2 hour	3 hour
69-State	100.38 ft	213.09 ft	415.98 ft	677.07 ft
13-State	108.89 ft	227.95 ft	326.04 ft	378.54 ft

4.5   Summary

This chapter was filled with a number of tables, filters and results from this research. For clarity, the following list summarizes these results.

4-29
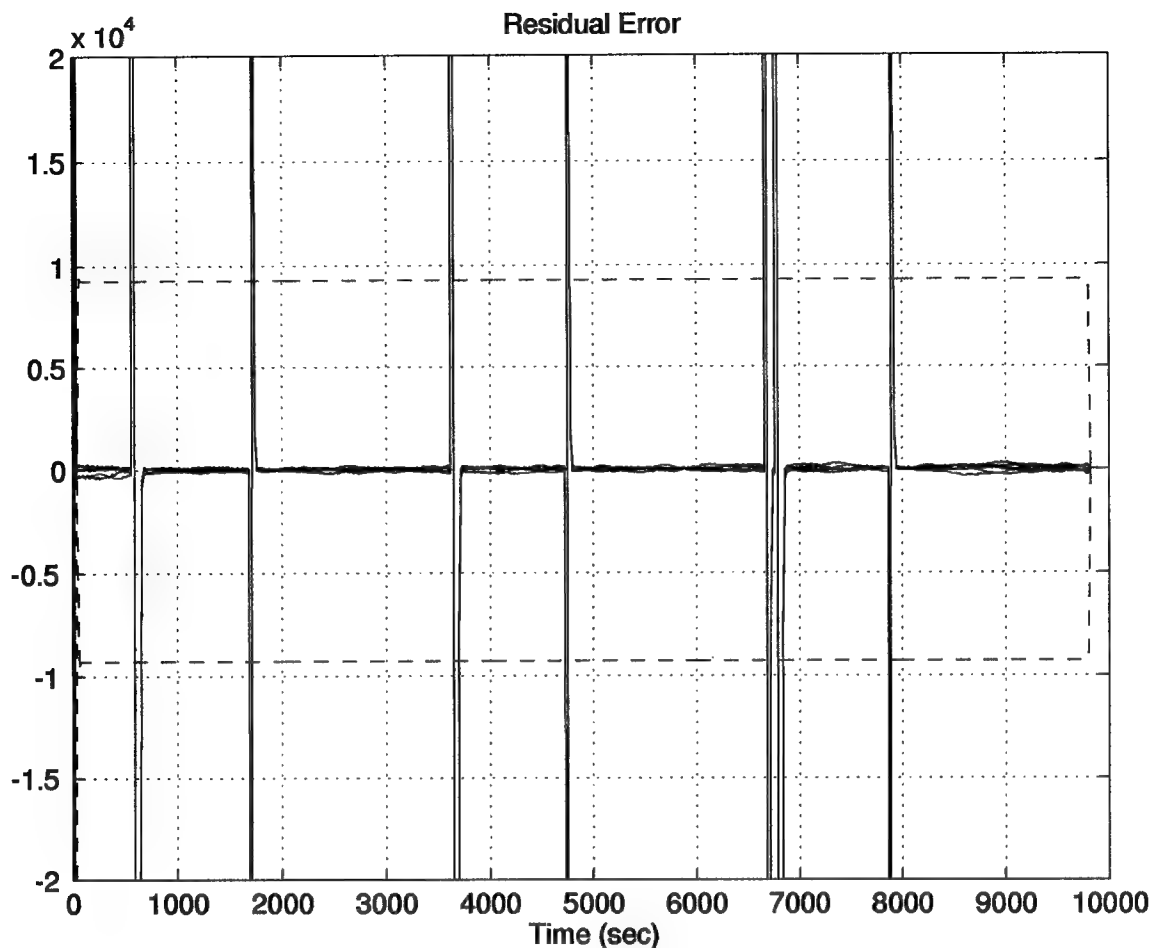
Figure 4.20   Residuals and Covariances of Residuals for the 69-State Filter

1. Computer Simulation: An alternative to MSOFE was developed for the purpose of simulating GPS/INS integrations. The simulation, MATSOFE, is easier from a developer's standpoint (7). Troubleshooting, model insertion, and plotting results were all advantages of MATSOFE. The major drawback to MATSOFE is computation speed. This slower computation time was not a detriment in this research because the measurement updates were accomplished once every 20 seconds. The update rate is the Achilles' heel of MATSOFE. MATSOFE slows down significantly as the measurement update rate is increased. MATSOFE is slow because MATLAB is not a compiled language.

2. Computer Simulation: The computer simulation did not require perfect Doppler radar measurements, as was necessary in past AFIT research (10, 18, 20, 22, 27).
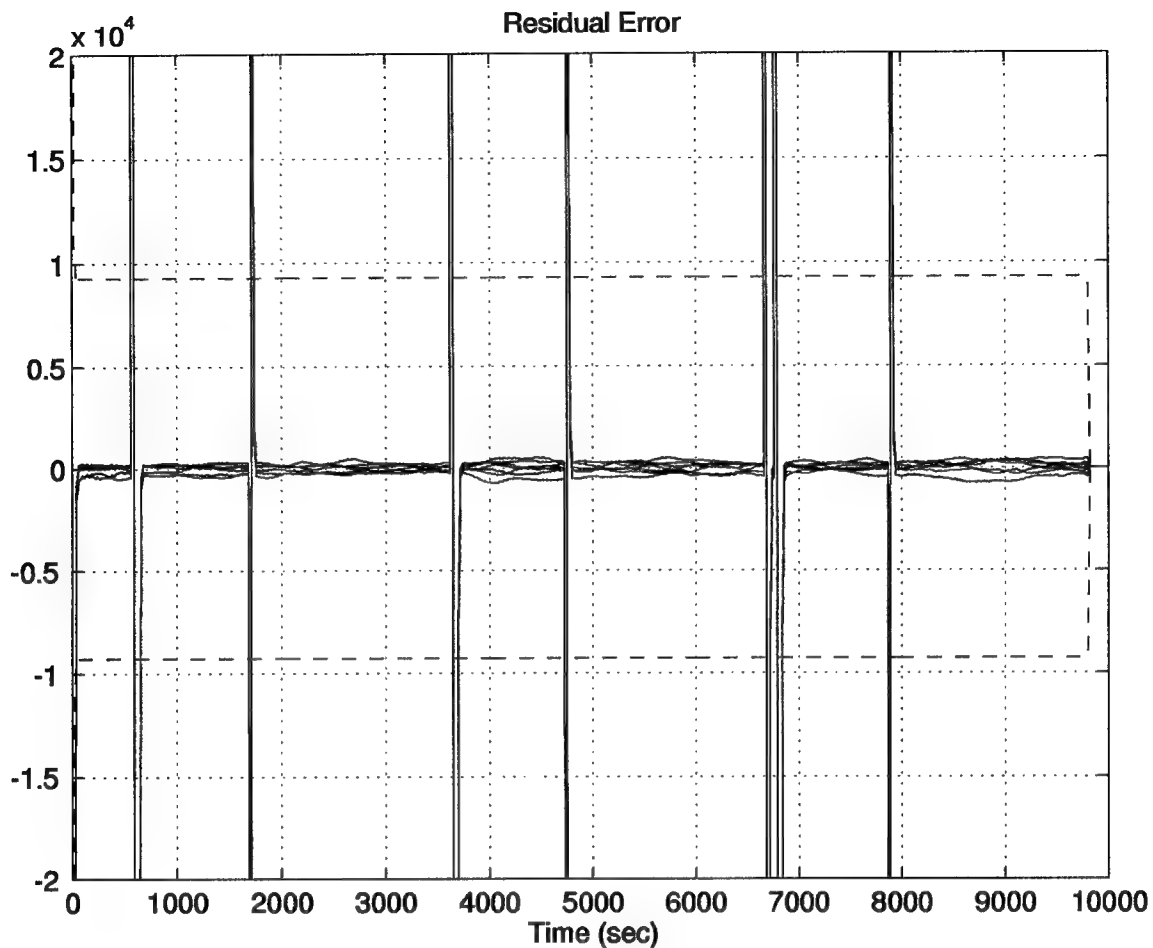
Figure 4.21   Residuals and Covariances of Residuals for the 13-State Filter

3. Computer Simulation: Simulations were run for 69-state and 13-state filters which were tuned with no knowledge of the hardware integration and then run again with the actual hardware integration tuning values. These results showed that there are differences between the real world GPS and INS systems and the simulations. For the most part, the simulations produced more optimistic results than the actual integration.

4. Models: The user clock bias and drift states in the GPS models were capable of being tuned to much smaller values in simulation (4 orders of magnitude) than is possible in real life. In fact, when the minute dynamic driving noise strength values determined from the simulation were placed into the actual GPS/INS integrated system, the filter would not even track.

5. Models: The 61-state filter was found to be the most accurate in the GPS/INS integrations. However, with only a slight loss in accuracy, the 49-state did remarkably well with 12 less states.

6. Models: The 41-state and 13-state filters were only three feet apart in temporally averaged mean error, yet the 41-state filter had 28 more INS error states.

7. Models: The 13-state filter had a large initial transient, most likely due to the reduction in vertical channel states, because the transient was not present in the 69-state filter. This transient died out after four to five updates.

8. Hardware Integration: The MATSOFE computer simulation was successfully modified to accept real data. These MATLAB m-files will now conduct a tightly-coupled GPS/INS integration in a post-processing environment.

9. Hardware Integration: Tuning is crucial. Nearly all of these results were tuned for a 20 second update period and a 1000 second time span. The EKF can be tuned for nearly any interval with varying results. For instance, when tuned for a shorter interval of time, the filter predictions are more accurate during that interval. As the interval increases, the nonlinearities due to inefficiencies in the model become more prevalent. The appropriate tuning then shifts the weighting toward the sensor measurements and away from the model.

10. Hardware Integration: A measurement update period of 10 seconds was 10 feet better in temporally averaged mean error over a 20 second update period; however, the 10 second update period is nearly twice as slow in computation.

11. Hardware Integration: The GPS/INS integration using the 13-state filter had the poorest results of the filters studied, yet even this filter was nearly twice as accurate as either GPS receiver alone and many times better than the INS alone.

12. Hardware Integration: Filters tuned for a 1000 second run and observed over a three-hour period revealed that the 13-state outperformed the 69-state filter. The 69-state filter model could not be tuned to provide a more accurate position estimate than the 13-state filter. These results suggest that the 69-state filter may be overmodeled for longer duration runs.

V. Conclusions and Recommendations

This chapter is divided into two sections: conclusions drawn from the results presented in Chapter 4 and recommendations for future AFIT research. The conclusions section generalizes the results of the computer simulation, the model analysis, and the hardware integration. The recommendations section points out potential problem areas identified in the research, provides suggestions to remedy these shortcomings, and recommends future topics to be included in future AFIT theses.

5.1 Conclusions

Several conclusions resulted from the research presented in Chapter IV. First, the computer simulations used at AFIT produce better position results than can be realized in the hardware integration of a tightly-coupled GPS and INS. The GPS and INS system truth models closely represent the system errors but, as expected, are not perfect. The deviations of the system truth models from the real world produce the errors in the computer simulations. When tuning future simulations, researchers should keep in mind that the user clock and drift states' dynamic driving noise strength values are very large in reality.

Five different filter models were evaluated in the hardware integration. The research determined that the current 69-state system truth model does not follow the actual errors as well as some of the other filters. The 61-state model represented the errors of the integrated system slightly better than the 69-state model. The 49-state model also outperformed the 69-state model and is obviously less burdensome to the computer simulation due to its reduced number of states.

The feedforward implementation provided very good results. The 13-state filter compared well to the larger state-dimensioned filters by providing slightly less accuracy at a fraction of the computation time for runs less than one hour. In addition, the 13-state filter outperformed the 69-state filter in runs exceeding one hour, which may indicate that the 69-state filter system overmodels the integrated system for longer term simulations. Tuning was critical in the accuracy of various filter models. The 13-state filter used in a feedforward implementation could be used to provide improved navigation performance in runs lasting up to three hours; however, the filter follows the latitude and longitude errors of the actual system best during the first 100 seconds of a run. The stabilization of the vertical channel in the INS prevents drifting.

Accuracy was improved by increasing the measurement update frequency from one update every 20 seconds to an update every 10 seconds. The temporally averaged mean error of the position estimate was reduced by 10 feet. With the computation power currently available at AFIT, this increase in update frequency will not prevent the implementation of a mobile GPS/INS integrated system. Computations on the 486 laptop computer with any of the filter models are possible with a real-time scenario at a 10-second update period with the advantage of improved performance. If an even greater measurement update frequency is required, then only the 13-state filter can be used in the integration due to computational restraints.

5.2 Recommendations

The following section provides this researcher's recommendations for future AFIT research topics, enhancements to the MatSOFE subroutines, and requirements to accomplishing the real-time integrated GPS/INS system.

5.2.1 Remodeling the User Clock and Drift States. In order to produce more realistic results, one could add more system truth noise strength to the user clock and drift states. In the current simulation system truth model, no system truth model noise is added to the user clock error states. The additional system truth model noise may help the simulations provide more realistic results. By adding system truth noise to the model, the filter dynamic driving noise must increase in magnitude.

5.2.2 Establishing the Size of the Observation Matrix. In order for AFIT to develop a real-time system, several issues must be resolved. The first issue is determining the order of the observation matrix, H, in the EKF. Currently, the subroutines read in the number of satellites in view and establish the size of the H matrix. The problem is created when the number of satellites changes during a run. The number of satellites will change several times during a one-hour run. In its current configuration, this fluctuating size of the H matrix causes an error in the routine: this is a coding error, not a conceptual error. All data files used in this research were 1000-second excerpts of longer data files. These 1000-second interval data files contained the same number of satellites throughout the duration of the run.

Two options exist for solving this problem. The easiest of these options is to set the observation matrix to accept a fixed number of satellites; either four or five satellites is realistic. All of the collected data had as a minimum five satellites and usually seven satellites were tracked. The benefit of this option is that longer portions of the data files could be more easily fed into the EKF. The drawback is that the fewer satellites, the higher the GDOP, and the less accurate the solutions. Although the algorithm may be simpler, this method throws out data and is not as desirable as the second option. During the run an algorithm could be written to select the satellites which provide the best GDOP. In those rare cases when there were less satellites than the preselected number, the algorithm could propagate the error states until another measurement update is possible.

Another option would be to change the m-files to accept all satellites in view. This option would be difficult to implement but would provide the best possible GDOPs available. In order to implement the second option in a real-time scenario, the EKF could be reset each time the number of satellites in view changes. In other words, every time the number of satellites changes the algorithm would readjust the matrix sizes. This adjustment would set the covariances back to their initial values and the H matrix could be reinitialized. The major disadvantage of this option is the initial transients present in the altitude states of the 13-state filter. Each time the satellite number changed and the EKF restarted, the transients would appear in the altitude estimates. This varying order of the H matrix would also present problems in the output of the residuals. For instance, some intervals of the run would have five sets of residuals, others six sets, and still yet some would have seven. This varying number of outputs would be difficult to output.

5.2.3 User Clock Biases.   As mentioned in Chapter IV, the XR-4PC had noticeable jumps in the user clock. These jumps were rather commonplace in occurrence, but their magnitudes varied rather drastically. Of course, these biases can be counteracted by increasing the Q in the clock bias and drift states. Since the jumps were not noticed in the XR-5PC data, this indicates there will be different sets of tuning values for each respective receiver. Further study is required, but it is very possible that a common GPS model is not possible. Just as various inertial systems have different truth models, the same would be expected with GPS receivers. In a GPS truth model, the satellite-related errors will be the same throughout all of the models. But as was demonstrated in this thesis, the user clock is a critical error state that will vary between GPS receiver models and quite possibly from receivers of the same type.

5.2.4   Access to Real-Time GPS Raw Data.   In order to implement a real-time and mobile integrated GPS/INS system, a method of obtaining the necessary information from the XR-4PC or XR-5PC must be devised. Currently, the GPS receivers in the laboratory will only write data to a file. In order for this information to be useful in integrating a GPS and INS in real-time, the GPS receivers must transmit the satellite ephemeris and pseudorange information over some sort of data bus. The LN-94 INS uses the 1553 data bus.

5.2.5   Real-Time Kalman Filtering.   Several of the Kalman filtering equations in the MATSOFE routines must be rewritten into the C programming language for the data bus controller. If this rehosting of the equations is accomplished, then real-time Kalman filtering can be done on the computer acting as the controller for the LN-94. The LN-94 passes information and accepts data over the 1553 data bus. The software which interfaces the 1553 data bus with the PC is known as DT-1120. The DT-1120 software can call out the user's subroutines written in C. Therefore, in a feedback implementation, the data monitor would read the LN-94's output file just as was done in this thesis. The DT-1120 software would then call out the Kalman filtering subroutines. The new position estimates would be sent to the screen or an output file. At a predetermined interval, the DT-1120 software would also send resets back to the INS from the EKF produced estimates. The

5.2.6   Real-Time Filter Model.   As discussed in Chapter 4, the 13-state filter had a large initial transient in the vertical channel. Instead of the 13-state filter model used in this research, a filter of increased order may produce better results. By adding more vertical channel states to the filter, the transient magnitude may be diminished. Recall that the filter models using the 39-state INS models did not exhibit altitude-related transients.

5.2.7   Research Topics.   Several recommended research topics come to mind for future AFIT students in order to accomplish the goal of a mobile real-time system. As a first step, a loosely-coupled GPS/INS integration could be attempted and compared to the tightly-coupled results of this thesis. Once the appropriate models are determined, the conversion of MATSOFE to accomplish a loosely-coupled integration should not be difficult. If the instability dilemma of filters driving filters can be resolved, a loosely-coupled integration may be more desirable for a real-time GPS/INS integration. A loosely-coupled system would seemingly require less computation time

in the EKF software; the satellite ECEF position computations would not be required. However, these satellite ECEF computations must still be accomplished in the GPS receiver's software.

Another important step toward a mobile GPS/INS integration is to conduct a real-time feedback integration of the GPS and INS. This research should concentrate on how often the INS must be reset and an analysis of the results versus the feedforward approach. From Figure 4.13, it would seem that the feedback implementation will provide much better position estimates and the INS will only be required to be reset every 100 to 200 seconds. Of course, the actual reset times will be contingent upon the position accuracies required from the integrated system. The type of integration would be a combination of the feedforward and feedback approaches discussed in Chapter 1 and shown in Figures 1.1 and 1.2. The feedback approach shown in Figure 1.2 resets the INS each time the EKF sends a new position estimate. The hybrid integration scheme in Figure 5.1 operates as a feedforward implementation and occasionally sends a reset to the INS in a feedback approach.
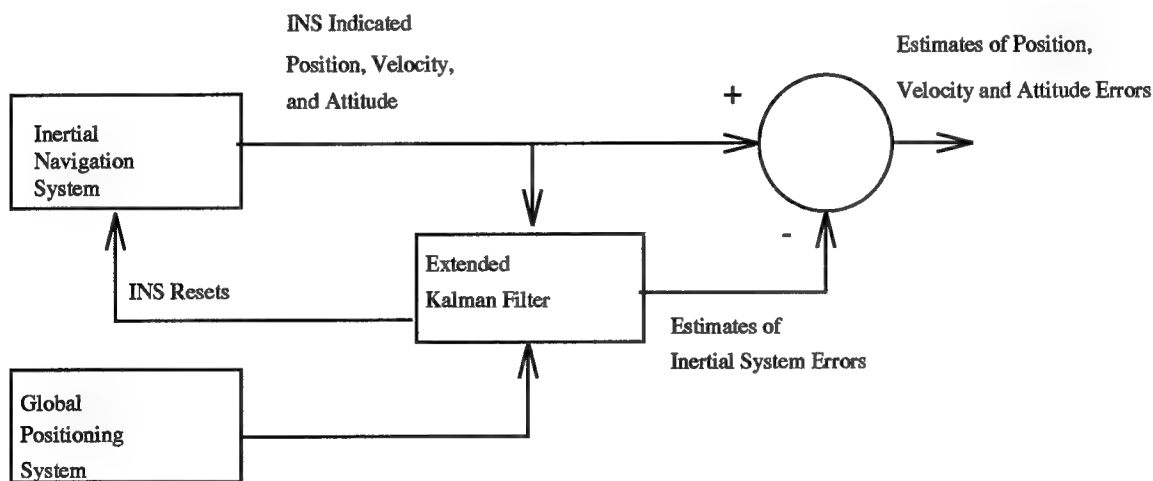
Figure 5.1   EKF Feedback Implementation for a GPS/INS Integration

A final area of recommended study is fault detection, isolation, and recovery (FDIR). FDIR studies have been conducted in simulation at AFIT (18, 27). FDIR research on the actual hardware integration should be conducted. This analysis could be conducted with the data files used in this research. Since the computations are done in a post-processing environment, the individual segments of the data files can be altered to induce failures. This research would mainly concentrate on satellite failures, but INS inconsistencies could also be induced.

Another area of concern in industry today is the incorrect resetting of the INS in a feedback configuration. The scheme illustrated in Figure 5.1 could be used to aid in developing an FDIR algorithm. The GPS-only and INS-only solutions could be compared to identify incorrect position estimates. The INS-only output would not be available in a true feedback configuration as shown in Figure 1.2, but would be available in Figure 5.1. Of course several factors would affect the results such as: tuning, interval of INS resets, measurement update rate, GPS satellite health word and the severity of vehicle dynamics.

5.3 Summary

Hopefully, the results of this research and the recommendations provided in this research will assist the AFIT Navigation, Guidance, and Control section in attaining its goal of developing a real-time integrated system. This chapter has presented the conclusions and recommendations culminating from this research.

Appendix A. Error State Models Definitions

This appendix contains tabular listings of the 93-state, 39-state, 11-state LN-93 INS models, and the 30-state, and 22-state GPS models.

Table A.1 93-state LN-93 INS Model, Category I: General Errors

State Number	State Symbol	Definition
1	$\delta\Theta_X$	X component of vector angle from true to computer frame
2	$\delta\Theta_Y$	Y component of vector angle from true to computer frame
3	$\delta\Theta_Z$	Z component of vector angle from true to computer frame
4	ϕ_X	X component of vector angle from true to platform frame
5	ϕ_Y	Y component of vector angle from true to platform frame
6	ϕ_Z	Z component of vector angle from true to platform frame
7	δV_X	X component of error in computer velocity
8	δV_Y	Y component of error in computer velocity
9	δV_Z	Z component of error in computer velocity
10	δh	Error in vehicle altitude above reference ellipsoid
11	δh_L	Error in lagged inertial altitude
12	δS_3	Error in vertical channel aiding state
13	δS_4	Error in vertical channel aiding state

Table A.2   93-state LN-93 INS Model, Category II: First Order Markov Process Error States

State Number	State Symbol	Definition
14	b_{Xc}	X component of gyro correlated drift rate
15	b_{Yc}	Y component of gyro correlated drift rate
16	b_{Zc}	Z component of gyro correlated drift rate
17	∇_{Xc}	X component of accelerometer & velocity quantizer correlated noise
18	∇_{Yc}	Y component of accelerometer & velocity quantizer correlated noise
19	∇_{Zc}	Z component of accelerometer & velocity quantizer correlated noise
20	δg_X	X component of gravity vector errors
21	δg_Y	Y component of gravity vector errors
22	δg_Z	Z component of gravity vector errors
23	δh_B	Total baro-altimeter correlated error
24	b_{Xt}	X component of gyro trend
25	b_{Yt}	Y component of gyro trend
26	b_{Zt}	Z component of gyro trend
27	∇_{Xc}	X component of accelerometer trend
28	∇_{Yc}	Y component of accelerometer trend
29	∇_{Zc}	Z component of accelerometer trend
30	b_X	X component of gyro drift rate repeatability
31	b_Y	Y component of gyro drift rate repeatability
32	b_Z	Z component of gyro drift rate repeatability
33	S_{gx}	X component of gyro scale factor error
34	S_{gy}	Y component of gyro scale factor error
35	S_{gz}	Z component of gyro scale factor error
36	χ_1	X gyro misalignment about Y axis
37	χ_2	Y gyro misalignment about X axis
38	χ_3	Z gyro misalignment about X axis
39	ν_1	X gyro misalignment about Z axis
40	ν_2	Y gyro misalignment about Z axis
41	ν_3	Z gyro misalignment about Y axis

Table A.3 93-state LN-93 INS Model, Category III: Gyro Bias Error States

State Number	State Symbol	Definition
42	D_{XXX}	X gyro scale factor non-linearity
43	D_{YYY}	Y gyro scale factor non-linearity
44	D_{ZZZ}	Z gyro scale factor non-linearity
45	S_{Qbx}	X gyro scale factor asymmetry error
46	S_{Qby}	Y gyro scale factor asymmetry error
47	S_{Qbz}	Z gyro scale factor asymmetry error

Table A.4 93-state LN-93 INS Model, Category IV: Accelerometer Bias Error States

State Number	State Symbol	Definition
48	∇_{bx}	X component of accelerometer bias repeatability
49	∇_{by}	Y component of accelerometer bias repeatability
50	∇_{bz}	Z component of accelerometer bias repeatability
51	S_{Ax}	X component of accelerometer & velocity quantizer scale factor error
52	S_{Ay}	Y component of accelerometer & velocity quantizer scale factor error
53	S_{Az}	Z component of accelerometer & velocity quantizer scale factor error
54	S_{QAx}	X component of accelerometer & velocity quantizer scale factor asymmetry
55	S_{QAy}	Y component of accelerometer & velocity quantizer scale factor asymmetry
56	S_{QAz}	Z component of accelerometer & velocity quantizer scale factor asymmetry
57	f_{XX}	Coefficient of error proportional to square of measured acceleration
58	f_{YY}	Coefficient of error proportional to square of measured acceleration
59	f_{ZZ}	Coefficient of error proportional to square of measured acceleration
60	f_{XY}	Coefficient of error proportional to products of acceleration along & orthogonal to accelerometer sensitive axis
61	f_{XZ}	Coefficient of error proportional to products of acceleration along & orthogonal to accelerometer sensitive axis
62	f_{YX}	Coefficient of error proportional to products of acceleration along & orthogonal to accelerometer sensitive axis
63	f_{YZ}	Coefficient of error proportional to products of acceleration along & orthogonal to accelerometer sensitive axis
64	f_{ZX}	Coefficient of error proportional to products of acceleration along & orthogonal to accelerometer sensitive axis
65	f_{ZY}	Coefficient of error proportional to products of acceleration along & orthogonal to accelerometer sensitive axis
66	μ_1	X accelerometer misalignment about Z axis
67	μ_2	Y accelerometer misalignment about Z axis
68	μ_3	Z accelerometer misalignment about Y axis
69	σ_3	Z accelerometer misalignment about X axis

Table A.5   93-state LN-93 INS Model, Category V: Thermal Transient Error States

State Number	State Symbol	Definition
70	∇_{X_q}	X component fo accelerometer bias thermal transient
71	∇_{Y_q}	Y component fo accelerometer bias thermal transient
72	∇_{Z_q}	Z component fo accelerometer bias thermal transient
73	b_{X_q}	X component of initial gyro drift rate bias thermal transient
74	b_{Y_q}	Y component of initial gyro drift rate bias thermal transient
75	b_{Z_q}	Z component of initial gyro drift rate bias thermal transient

Table A.6   93-state LN-93 INS Model, Category VI: Gyro Compliance Error States

State Number	State Symbol	Definition
76	F_{XYZ}	X gyro compliance term
77	F_{XYY}	X gyro compliance term
78	F_{XYX}	X gyro compliance term
79	F_{XZY}	X gyro compliance term
80	F_{XZZ}	X gyro compliance term
81	F_{XZX}	X gyro compliance term
82	F_{YZX}	Y gyro compliance term
83	F_{YZZ}	Y gyro compliance term
84	F_{YZY}	Y gyro compliance term
85	F_{YXZ}	Y gyro compliance term
86	F_{YXX}	Y gyro compliance term
87	F_{YXY}	Y gyro compliance term
88	F_{ZXY}	Z gyro compliance term
89	F_{ZXX}	Z gyro compliance term
90	F_{ZXZ}	Z gyro compliance term
92	F_{ZYX}	Z gyro compliance term
93	F_{ZYY}	Z gyro compliance term
93	F_{ZYZ}	Z gyro compliance term

Table A.7   39-state LN-93 INS Model, First 24 States

State Number	State Symbol	Definition	LN-93 State
1	$\delta\Theta_X$	X component of vector angle from true to computer frame	1
2	$\delta\Theta_Y$	Y component of vector angle from true to computer frame	2
3	$\delta\Theta_Z$	Z component of vector angle from true to computer frame	3
4	ϕ_X	X component of vector angle from true to platform frame	4
5	ϕ_Y	Y component of vector angle from true to platform frame	5
6	ϕ_Z	Z component of vector angle from true to platform frame	6
7	δV_X	X component of error in computer velocity	7
8	δV_Y	Y component of error in computer velocity	8
9	δV_Z	Z component of error in computer velocity	9
10	δh	Error in vehicle altitude above reference ellipsoid	10
11	δh_L	Error in lagged inertial altitude	23
14	δS_3	Error in vertical channel aiding state	11
15	δS_4	Error in vertical channel aiding state	12
16	δS_4	Error in vertical channel aiding state	13
17	δS_4	Error in vertical channel aiding state	17
18	δS_4	Error in vertical channel aiding state	18
19	δS_4	Error in vertical channel aiding state	19
20	δS_4	Error in vertical channel aiding state	20
21	δS_4	Error in vertical channel aiding state	21
22	δS_4	Error in vertical channel aiding state	22

Table A.8   39-state LN-93 INS Model, Second 19 States

State Number	State Symbol	Definition	LN-93 State
23	$\delta\Theta_X$	X component of vector angle from true to computer frame	30
24	$\delta\Theta_Y$	Y component of vector angle from true to computer frame	31
25	$\delta\Theta_Z$	Z component of vector angle from true to computer frame	32
26	ϕ_X	X component of vector angle from true to platform frame	33
27	ϕ_Y	Y component of vector angle from true to platform frame	34
28	ϕ_Z	Z component of vector angle from true to platform frame	35
29	δV_X	X component of error in computer velocity	48
30	δV_Y	Y component of error in computer velocity	49
31	δV_Z	Z component of error in computer velocity	50
32	δh	Error in vehicle altitude above reference ellipsoid	51
33	δh_L	Error in lagged inertial altitude	52
34	δS_3	Error in vertical channel aiding state	53
35	δS_4	Error in vertical channel aiding state	54
36	δS_4	Error in vertical channel aiding state	55
37	δS_4	Error in vertical channel aiding state	56
38	δS_4	Error in vertical channel aiding state	66
39	δS_4	Error in vertical channel aiding state	67
40	δS_4	Error in vertical channel aiding state	68
41	δS_4	Error in vertical channel aiding state	69

Table A.9   11-state LN-93 INS Model

State Number	State Symbol	Definition
1	$\delta\Theta_X$	X component of vector angle from true to computer frame
2	$\delta\Theta_Y$	Y component of vector angle from true to computer frame
3	$\delta\Theta_Z$	Z component of vector angle from true to computer frame
4	ϕ_X	X component of vector angle from true to platform frame
5	ϕ_Y	Y component of vector angle from true to platform frame
6	ϕ_Z	Z component of vector angle from true to platform frame
7	δV_X	X component of error in computer velocity
8	δV_Y	Y component of error in computer velocity
9	δV_Z	Z component of error in computer velocity
10	δh	Error in vehicle altitude above reference ellipsoid
11	δh_L	Error in lagged inertial altitude

Table A.10   30-state GPS Error Model

State Number	State Symbol	Definition
1	δR_{uclk}	User clock bias
2	δD_{uclk}	User clock drift
3	δR_{cloop1}	SV 1 code loop error
4	δR_{trop1}	SV 1 tropospheric error
5	δR_{ion1}	SV 1 ionospheric
6	δR_{clk1}	SV clock error
7	δX_1	SV X 1 component of position error
8	δY_1	SV Y 1 component of position error
9	δZ_1	SV Z 1 component of position error
10	δR_{cloop2}	SV 2 code loop error
11	δR_{trop2}	SV 2 tropospheric error
12	δR_{ion2}	SV 2 ionospheric
13	δR_{clk2}	SV 2 clock error
14	δX_2	SV X 2 component of position error
15	δY_2	SV Y 2 component of position error
16	δZ_2	SV Z 2 component of position error
17	δR_{cloop3}	SV 3 code loop error
18	δR_{trop3}	SV 3 tropospheric error
19	δR_{ion3}	SV 3 ionospheric
20	δR_{clk3}	SV 3 clock error
21	δX_3	SV X 3 component of position error
22	δY_3	SV Y 3 component of position error
23	δZ_3	SV Z 3 component of position error
24	δR_{cloop4}	SV 4 code loop error
25	δR_{trop4}	SV 4 tropospheric error
26	δR_{ion4}	SV 4 ionospheric
27	δR_{clk4}	SV 4 clock error
28	δX_4	SV X 4 component of position error
29	δY_4	SV Y 4 component of position error
30	δZ_4	SV Z 4 component of position error

Table A.11 22-state GPS Error Model

State Number	State Symbol	Definition
1	δR_{uclk}	User clock bias
2	δD_{uclk}	User clock drift
3	δR_{cloop1}	SV 2 code loop error
4	δR_{trop1}	SV 1 tropospheric error
5	δR_{ion1}	SV 1 ionospheric
6	δR_{clk1}	SV clock error
7	δX_1	SV 1 line-of-sight position error
8	δR_{cloop2}	SV 2 code loop error
9	δR_{trop2}	SV 2 tropospheric error
10	δR_{ion2}	SV 2 ionospheric
11	δR_{clk2}	SV 2 clock error
12	δX_2	SV 2 line-of-sight position error
13	δR_{cloop3}	SV 3 code loop error
14	δR_{trop3}	SV 3 tropospheric error
15	δR_{ion3}	SV 3 ionospheric
16	δR_{clk3}	SV 3 clock error
17	δX_3	SV 3 line-of-sight position error
18	δR_{cloop4}	SV 4 code loop error
19	δR_{trop4}	SV 4 tropospheric error
20	δR_{ion4}	SV 4 ionospheric
21	δR_{clk4}	SV 4 clock error
22	δX_4	SV 4 line-of-sight position error

Table A.12 10-state GPS Error Model

State Number	State Symbol	Definition
1	δR_{uclk}	User clock bias
2	δD_{uclk}	User clock drift
3	δR_{rng1}	Satellite 1 range error
4	$\delta R_{rngrate1}$	Satellite 1 range-rate error
5	δR_{rng2}	Satellite 2 range error
6	$\delta R_{rngrate2}$	Satellite 2 range-rate error
7	δR_{rng3}	Satellite 3 range error
8	$\delta R_{rngrate3}$	Satellite 3 range-rate error
9	δR_{rng4}	Satellite 4 range error
10	$\delta R_{rngrate4}$	Satellite 4 range-rate error

Table A.13   13-state Reduced Order Filter Model

State Number	State Symbol	Definition
1	$\delta\Theta_X$	X component of vector angle from true to computer frame
2	$\delta\Theta_Y$	Y component of vector angle from true to computer frame
3	$\delta\Theta_Z$	Z component of vector angle from true to computer frame
4	ϕ_X	X component of vector angle from true to platform frame
5	ϕ_Y	Y component of vector angle from true to platform frame
6	ϕ_Z	Z component of vector angle from true to platform frame
7	δV_X	X component of error in computer velocity
8	δV_Y	Y component of error in computer velocity
9	δV_Z	Z component of error in computer velocity
10	δh	Error in vehicle altitude above reference ellipsoid
11	δh_L	Error in lagged inertial altitude
12	δx_{uclkb}	User clock bias
13	δx_{uclkd}	User clock drift

Appendix B. Dynamics Matrices

Chapter 3 describes the truth and filter model dynamics matrices. These matrices are broken down into submatrices: $F_{Filter}, F_{INS_{t1}}, F_{INS_{t2}}, and F_{GPS_t}$. The variables and their respective units are described in the upcoming tables and defined in the LN-93 Error Budget (8). The structure of the dynamics matrices below correspond to the system truth model state definitions in Appendix A.

Table B.1   Elements of the Dynamics Submatrix F_{Filter}

Element	Term	Element	Term
(1,3)	$-\rho_y$	(1,8)	$-C_{RY}$
(2,3)	ρ_x	(2,7)	C_{RX}
(3,1)	ρ_y	(3,2)	$-\rho_x$
(4,2)	$-\Omega_z$	(4,3)	Ω_y
(4,5)	ω_{itz}	(4,6)	ω_{ity}
(4,8)	$-C_{RY}$	(5,1)	Ω_z
(5,3)	$-\Omega_x$	(5,4)	$-\omega_{itz}$
(5,6)	$-\omega_x$	(5,7)	$-C_{RX}$
(6,1)	$-\Omega_y$	(6,2)	Ω_x
(6,4)	ω_{ity}	(6,5)	$-\omega_{itx}$
(7,1)	$-2V_y\Omega_y$	(7,2)	$2V_zC_{RX}$
(7,3)	$A2V_z\Omega_y$	(7,5)	$-A_z$
(7,6)	A_y	(7,7)	$-V_zC_{RX}$
(7,8)	$2\Omega_z$	(7,9)	$-\rho_y - 2\Omega_y$
(8,1)	$2V_x\Omega_y$	(8,2)	$-2V_x\Omega_x - 2V_z\Omega_z$
(8,3)	$2V_x\Omega_y$	(8,4)	A_z
(8,6)	$-A_x$	(8,7)	$-2\Omega_z$
(8,8)	$-V_zC_{RY}$	(8,9)	$\rho_x + 2\Omega_x$
(9,1)	$2V_x\Omega_z$	(9,2)	$2V_y\Omega_z$
(9,3)	$-2V_y\Omega_y - 2V_x\Omega_x$	(9,4)	$-A_y$
(9,5)	A_x	(9,7)	$\rho_y + 2\Omega_y + V_xC_{RX}$
(9,8)	$-\rho_x - 2\Omega_x + C_{RY}$	(9,10)	$2g_o/a$
(9,11)	$-k_2$	(10,9)	1
(10,11)	$-k_1$	(11,10)	1
(11,11)	-1	(12,13)	$1ft^2/sec$

Table B.2   Elements of the Dynamics Submatrix $F_{INS_{t1}}$

Element	Term	Element	Term	Element	Term
(4,23)	C_{11}	(4,24)	C_{12}(4,25)	C_{13}	
(9,26)	$C_{11}\omega_{ibx}$	(9,27)	$C_{12}\omega_{iby}$	(9,28)	$C_{13}\omega_{ibz}$
(5,23)	C_{21}	(5,24)	C_{22}	(5,25)	C_{23}
(5,26)	$C_{21}\omega_{ibx}$	(5,27)	$C_{22}\omega_{iby}$	(5,28)	$C_{23}\omega_{ibz}$
(6,23)	C_{31}	(6,24)	C_{32}	(6,25)	C_{33}
(6,26)	$C_{31}\omega_{ibx}$	(6,27)	$C_{32}\omega_{iby}$	(6,28)	$C_{33}\omega_{ibz}$
(7,16)	C_{11}	(7,17)	C_{12}	(7,18)	C_{13}
(7,19)	1	(7,29)	C_{11}	(7,30)	C_{12}
(7,31)	C_{31}	(7,32)	$C_{11}A_x^B$	(7,33)	$C_{12}A_y^B$
(7,34)	$C_{13}A_z^B$	(7,35)	$C_{11}\lvert A_x^B\rvert$	(7,36)	$C_{12}\lvert A_y^B\rvert-$
(7,37)	$C_{13}\lvert A_z^B\rvert$	(7,38)	$C_{11}A_y^B$	(7,39)	$-C_{12}A_x^B$
(7,40)	$C_{13}A_y^B$	(7,41)	$C_{13}A_x^B$	(8,16)	C_{21}
(8,17)	C_{22}	(8,18)	C_{23}	(8,20)	1
(8,29)	C_{21}	(8,30)	C_{22}	(8,31)	C_{23}
(8,32)	$C_{21}A_x^B$	(8,33)	$C_{22}A_y^B$	(8,34)	$C_{23}A_z^B$
(8,35)	$C_{21}\lvert A_x^B\rvert$	(8,36)	$C_{22}\lvert A_y^B\rvert$	(8,37)	$C_{23}\lvert A_z^B\rvert$
(8,38)	$C_{21}A_y^B$	(8,39)	$-C_{22}A_x^B$	(8,40)	$C_{23}A_y^B$
(8,41)	$C_{23}A_x^B$	(9,14)	-1	(9,15)	k_2
(9,16)	C_{31}	(9,17)	C_{32}	(9,18)	C_{33}
(9,21)	1	(9,22)	k_2	(9,29)	C_{31}
(9,30)	C_{32}	(9,31)	C_{33}	(9,32)	$C_{31}A_x^B$
(9,33)	$C_{32}A_y^B$	(9,34)	$C_{33}A_z^B$	(9,35)	$C_{32}\lvert A_y^B\rvert$
(9,36)	$C_{31}\lvert A_x^B\rvert$	(9,37)	$C_{33}\lvert A_z^B\rvert$	(9,38)	$C_{31}A_x^B$
(9,39)	$-C_{33}A_x^B$	(9,40)	$C_{33}A_y^B$	(10,15)	k_1-1
(14,11)	k_3	(14,15)	$-k_3$	(14,22)	$-k_3$
(15,10)	k_4	(15,11)	$-k_4$	(15,22)	$k_4/600$
(15,15)	k_4-1				

Table B.3 Elements of the Dynamics Submatrix $F_{INS_{t2}}$

Element	Term	Element	Term	Element	Term
(16,16)	$-\beta_{\nabla_{xc}}$	(17,17)	$-\beta_{\nabla_{yc}}$	(18,18)	$-\beta_{\nabla_{zc}}$
(19,19)	$-\beta_{deltagx}$	(20,20)	$-\beta_{deltagy}$	(21,21)	$-\beta_{deltagz}$

Table B.4 Elements of the Dynamics Submatrix F_{GPS}

Element	Term	Element	Term	Element	Term
(42,42)	$-1\,ft^2/sec$	(43,43)	$-1/500\,ft^2/sec$	(44,44)	$-1/1500\,ft^2/sec$
(49,49)	$-1\,ft^2/sec$	(50,50)	$-1/500\,ft^2/sec$	(51,51)	$-1/1500\,ft^2/sec$
(56,56)	$-1\,ft^2/sec$	(57,57)	$-1/500\,ft^2/sec$	(58,58)	$-1/1500\,ft^2/sec$
(63,63)	$-1\,ft^2/sec$	(64,64)	$-1/500\,ft^2/sec$	(65,65)	$-1/1500\,ft^2/sec$

Appendix C. Comparison of 13-State Filter Results Using INS States 23 and 11

C.1   Overview

As discussed in Chapter 3, state 11, *the error in the lagged inertial altitude*, produced better altitude estimates than state 23, *the total correlated baro-altimeter error*, when used in the 13-state filter model. A complete description of the INS filter model states are given in Appendix A. The following discussion presents the background and research results leading to this filter model change.

C.2   Background

Past AFIT research has employed state 23 in the 13-state filter model (10, 18, 20, 22, 27). Initially, this researcher used the same filter model in computer simulation with satisfactory results. Figure C.1 shows these results. After the the computer simulation research was completed, an actual hardware integration was conducted using real data with state 23 in the 13-state filter as described in Section 4.4.1. Tuning of the 13-state filter was found to be extremely difficult with extreme errors in the altitude channel. The EKF was extremely slow in recovering from the initial transient altitude error. Figure C.3 illustrates the altitude errors using real data in the EKF. Although many different tuning values were attempted, the altitude errors were on the order of hundreds of feet and the recovery from the transient was very slow.

In an effort to correct the altitude error problem, state 11 was inserted into the model in place of state 23. State 11 was chosen on the basis of its position in the Litton LN-93 93-state error model, using the assumption that the vertical channel states were placed in the Litton model in the order of their significance (8). Research of past AFIT theses (10, 18, 20, 22, 27) did not reveal the reasoning for use of state 23. However, all of these theses did use the barometric altimeter as a sensor measurement, while modeling it in the filter as well. The results using state 11 in lieu of state 23 are shown in Figures C.2 and C.4.

C.3   Results

The computer simulation results did not indicate a difference in estimating altitude errors. These results are shown in Figures C.1 and C.2. However, the results are easily seen when real data is used in a hardware integration. These results are shown in Figures C.3 and 4.4.
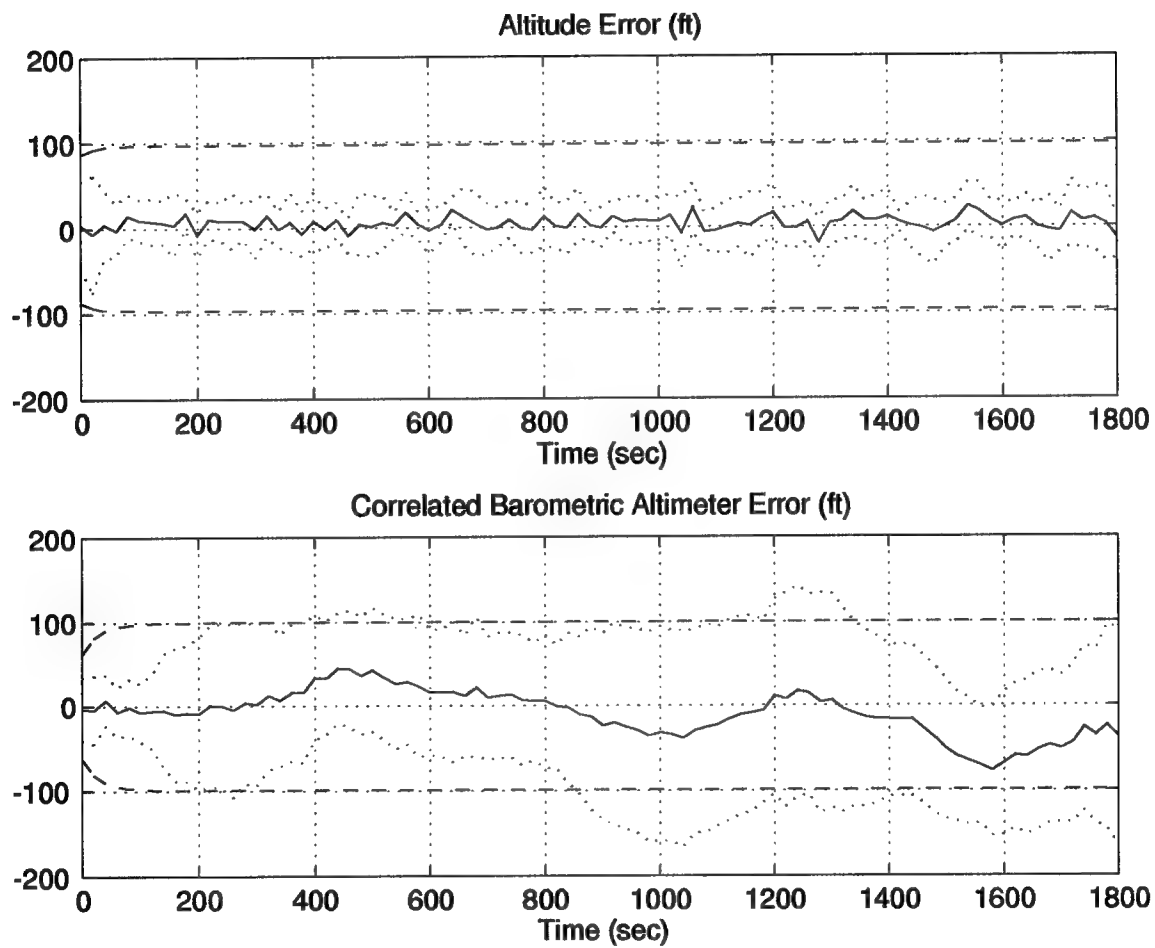
Figure C.1   Computer Simulation Altitude Errors Using Filter State 23
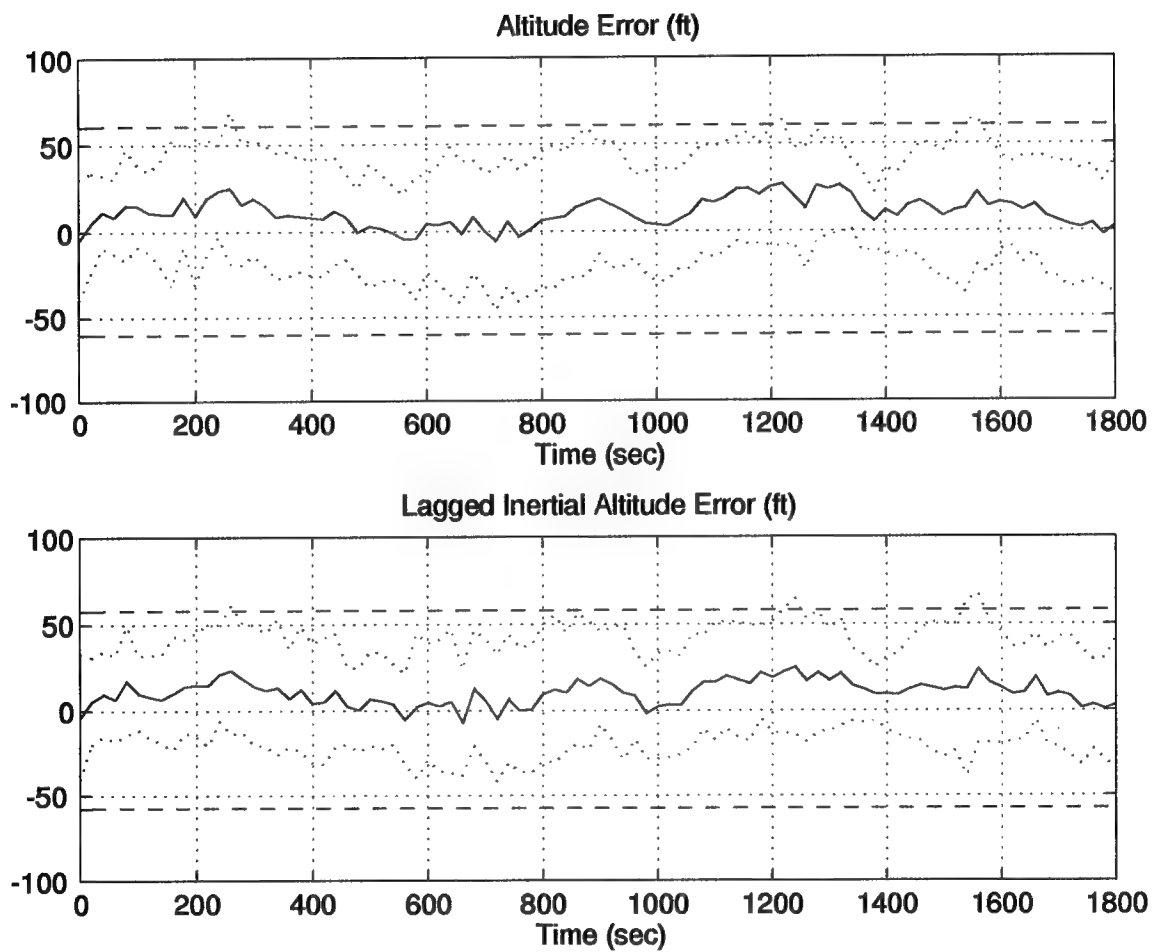
Figure C.2   Computer Simulation Altitude Errors Using Filter State 11
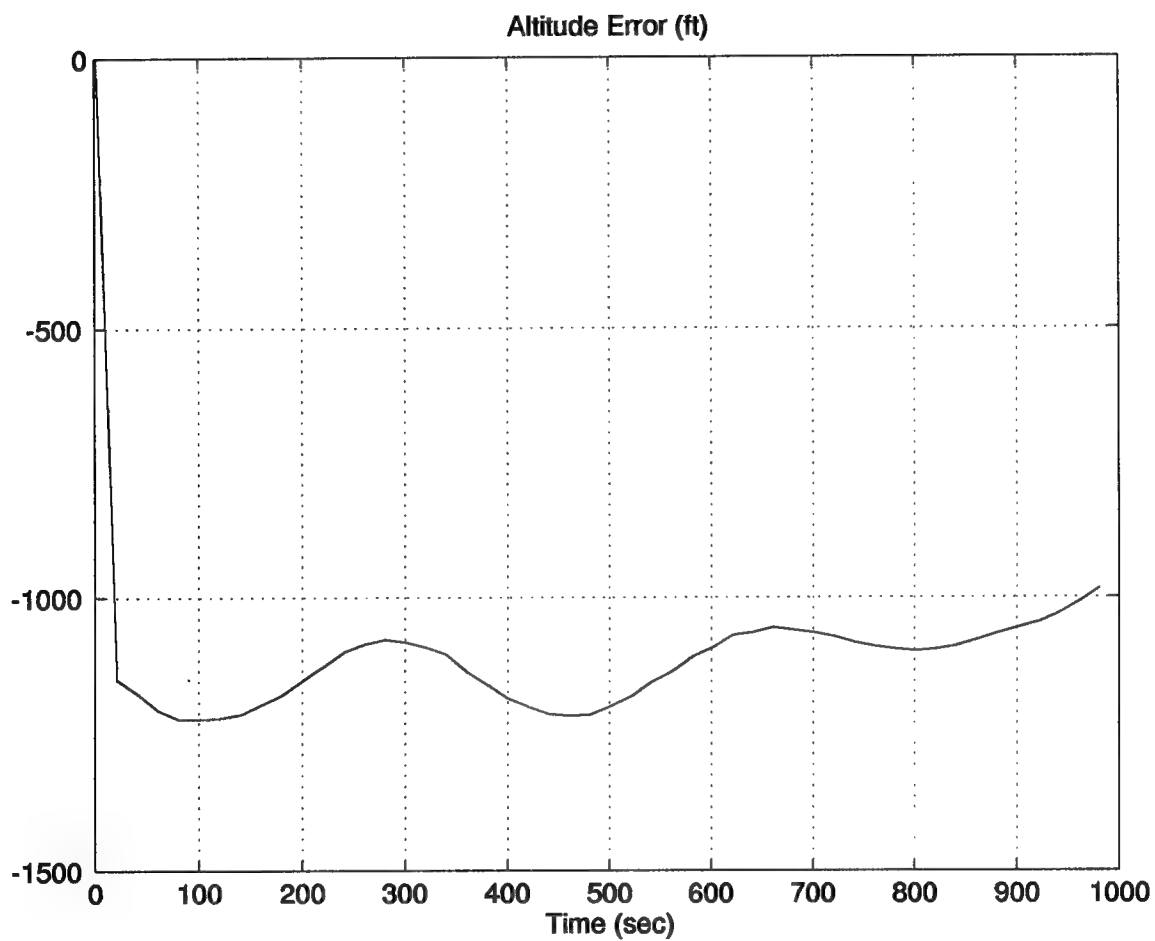
Figure C.3   Real Data Altitude Errors Using Filter State 23
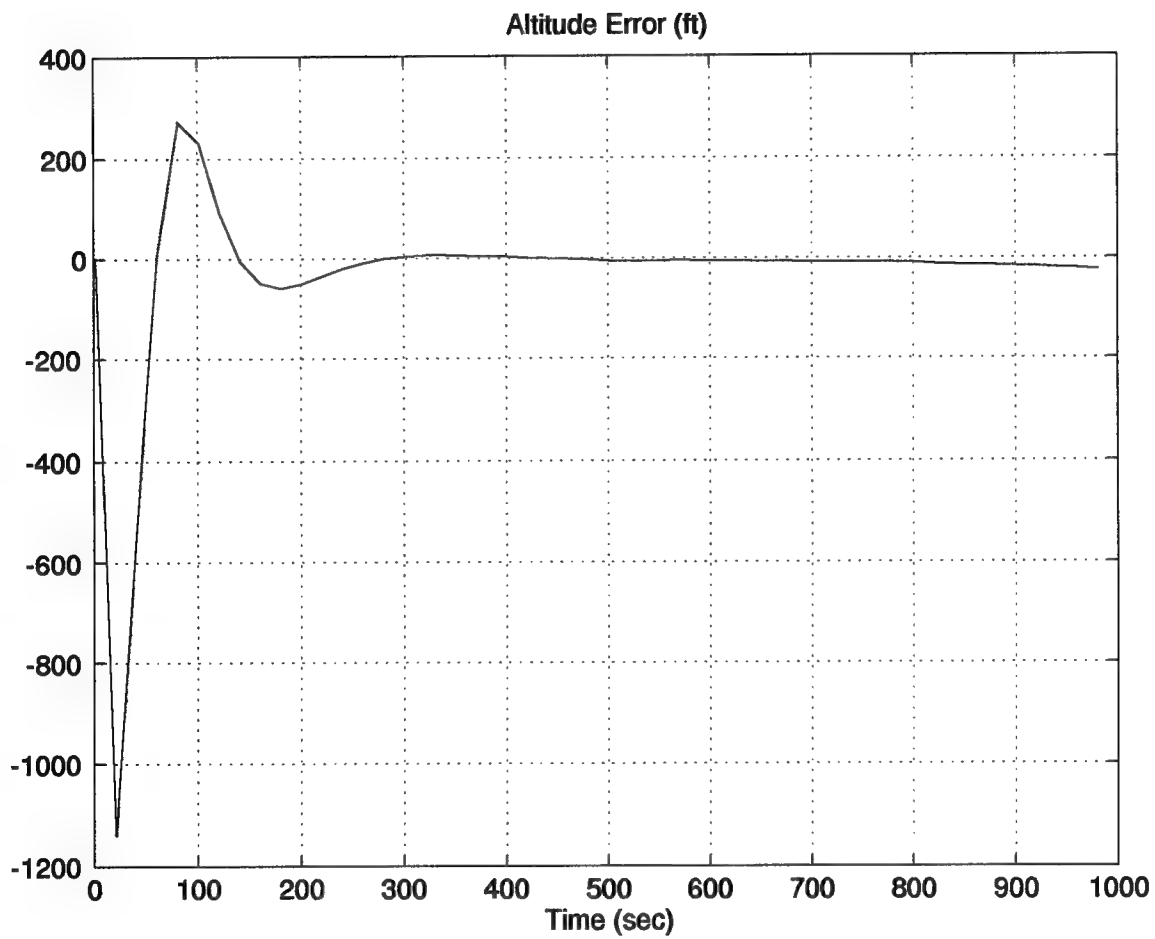
Figure C.4   Real Data Altitude Errors Using Filter State 11

Appendix D.   The 69-State Filter Category I Computer Simulation Results

The following plots represent the results of a 10-run Monte Carlo analysis using a 69-state filter. The tuning values were chosen without any knowledge of the actual hardware integration.

Table D.1   Legend of Filter Tuning Plots

| Symbol	Definition
Solid Line	Mean Error
Dotted Line	Mean Error ± True Sigma
Dashed Line	± Filter Predicted Sigma
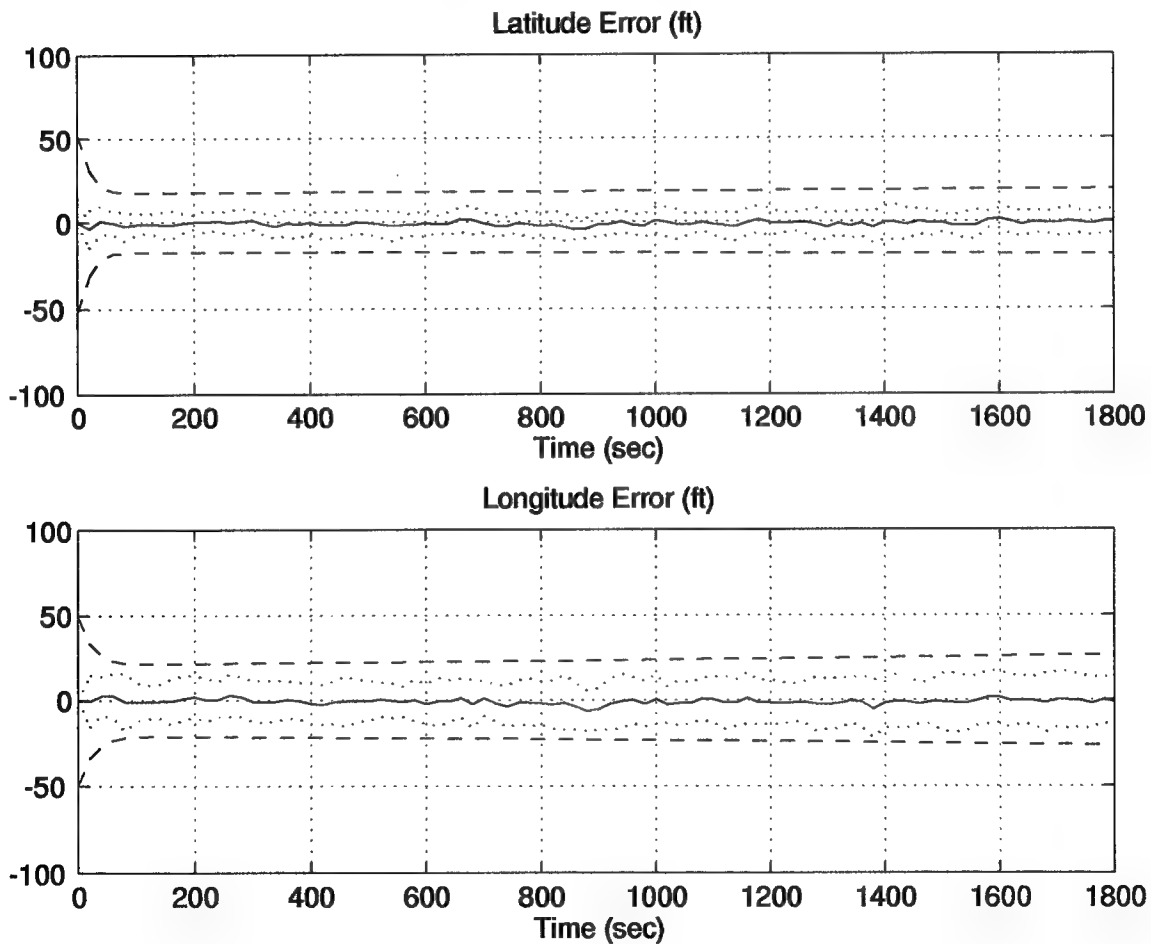
Figure D.1   Position Errors for the 69-State Filter Computer Simulation
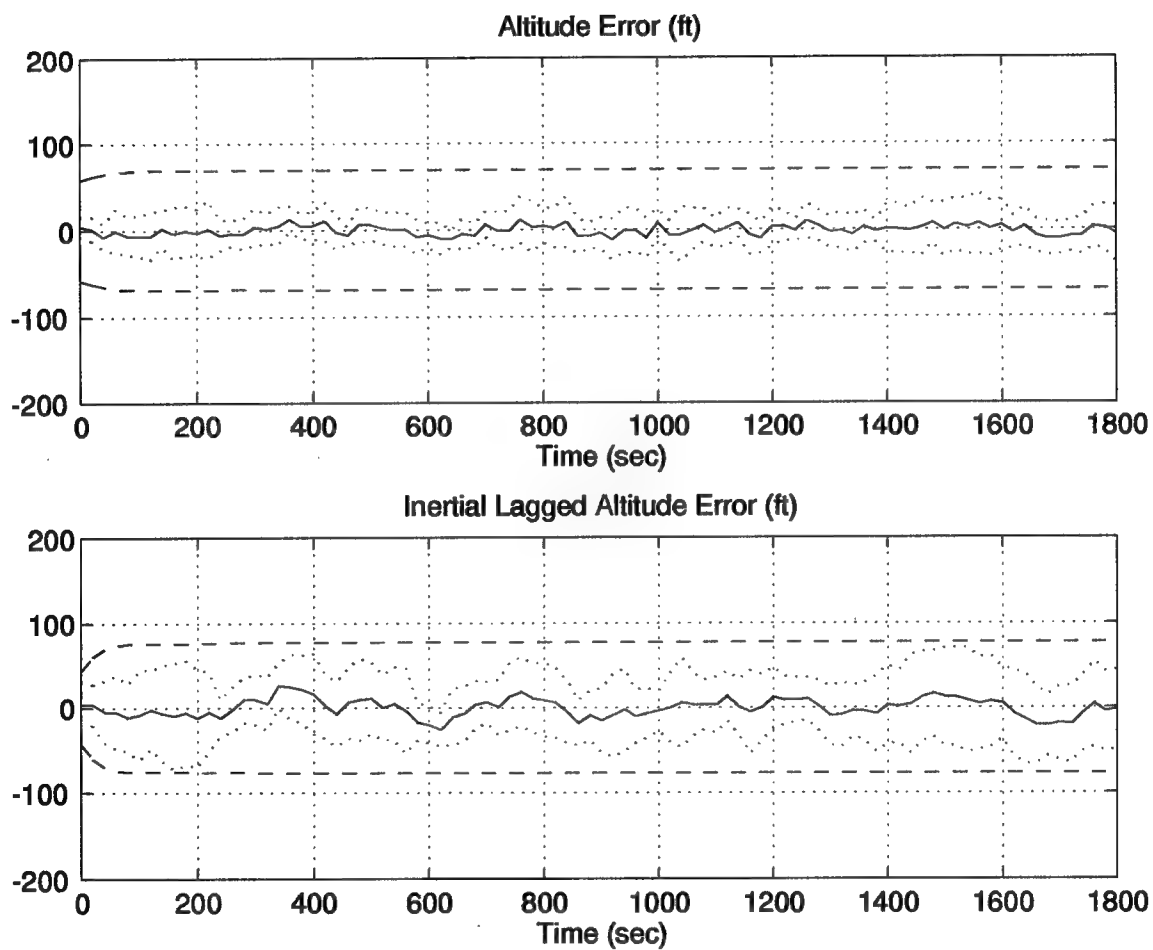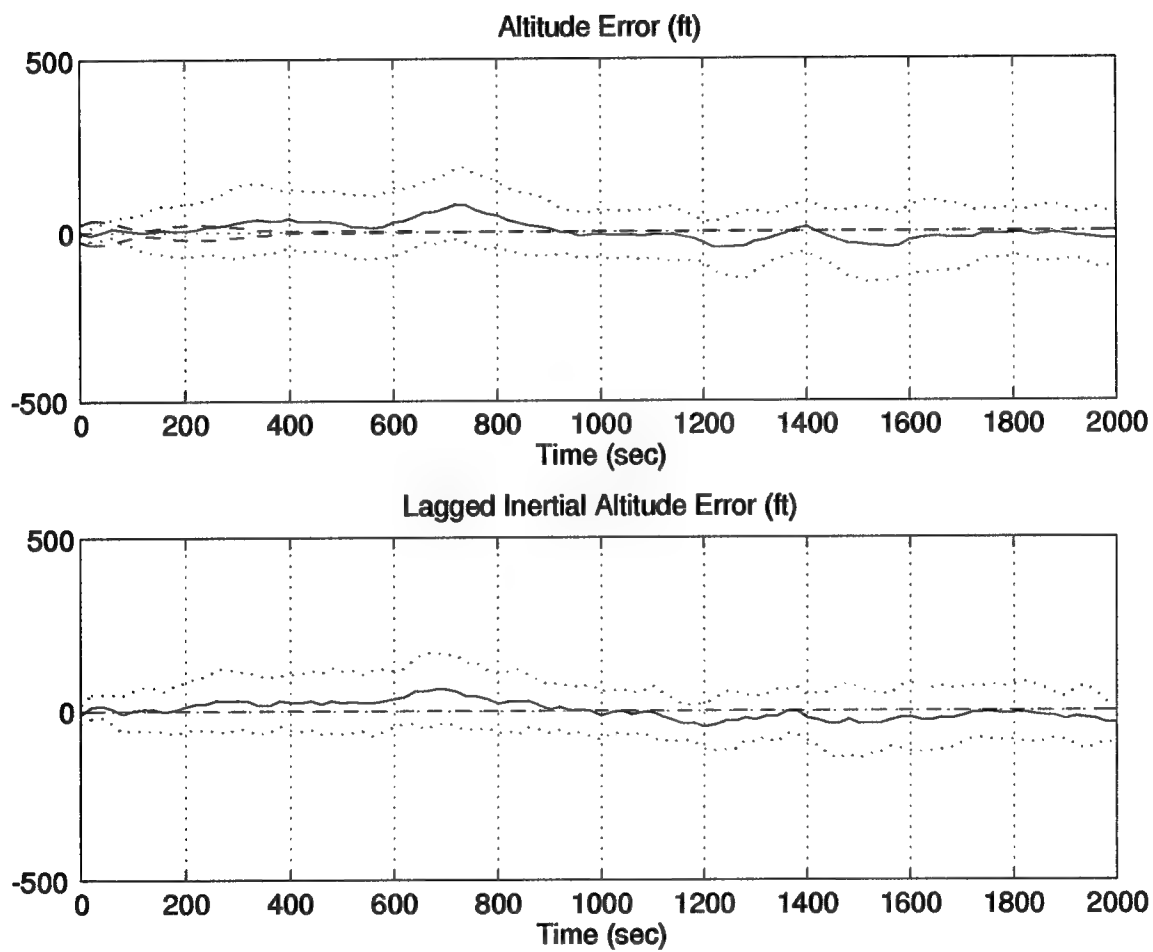
Figure D.2   Altitude Errors for the 69-State Filter Computer Simulation
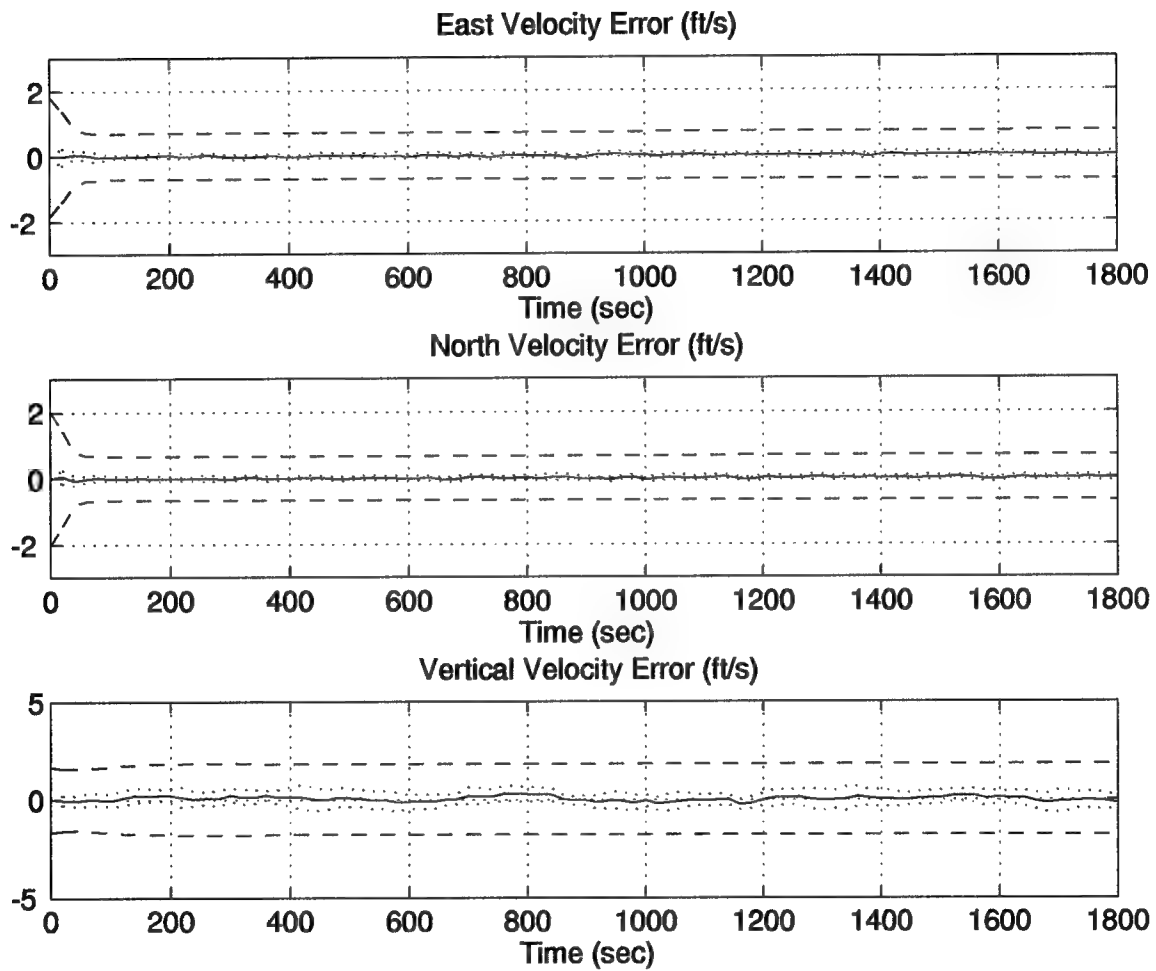
Figure D.3 Velocity Errors for the 69-State Filter Computer Simulation
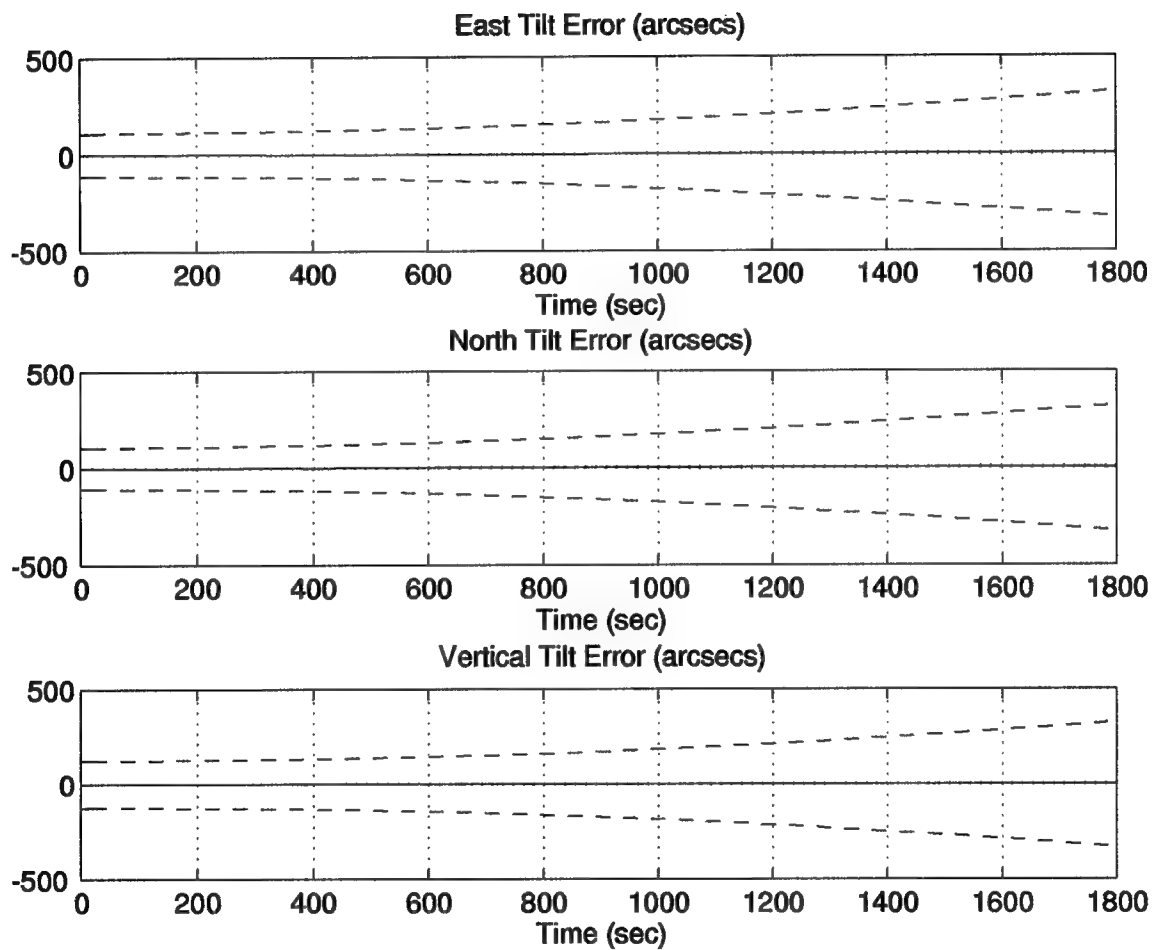
Figure D.4   Tilt Errors for the 69-State Filter Computer Simulation
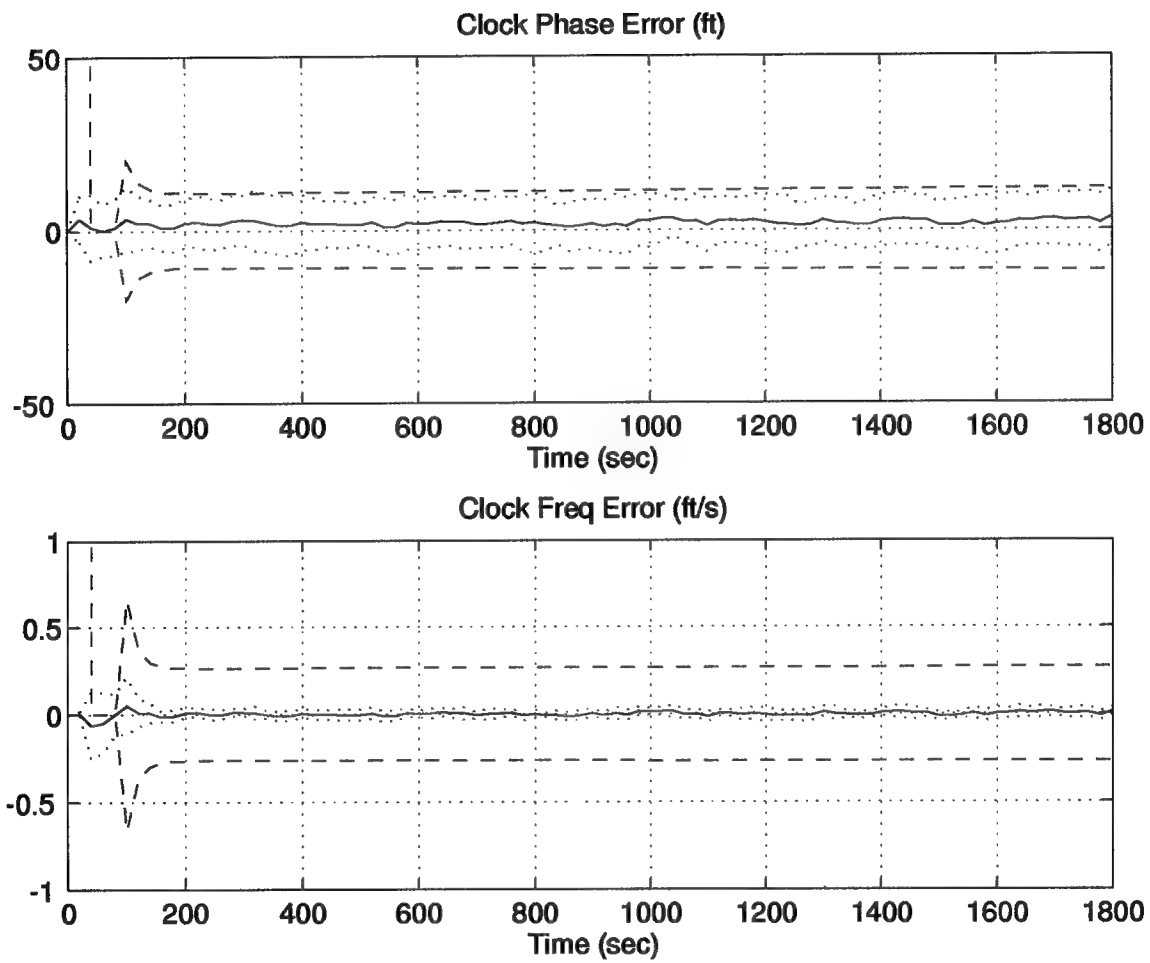
Figure D.5   Clock Errors for the 69-State Filter Computer Simulation
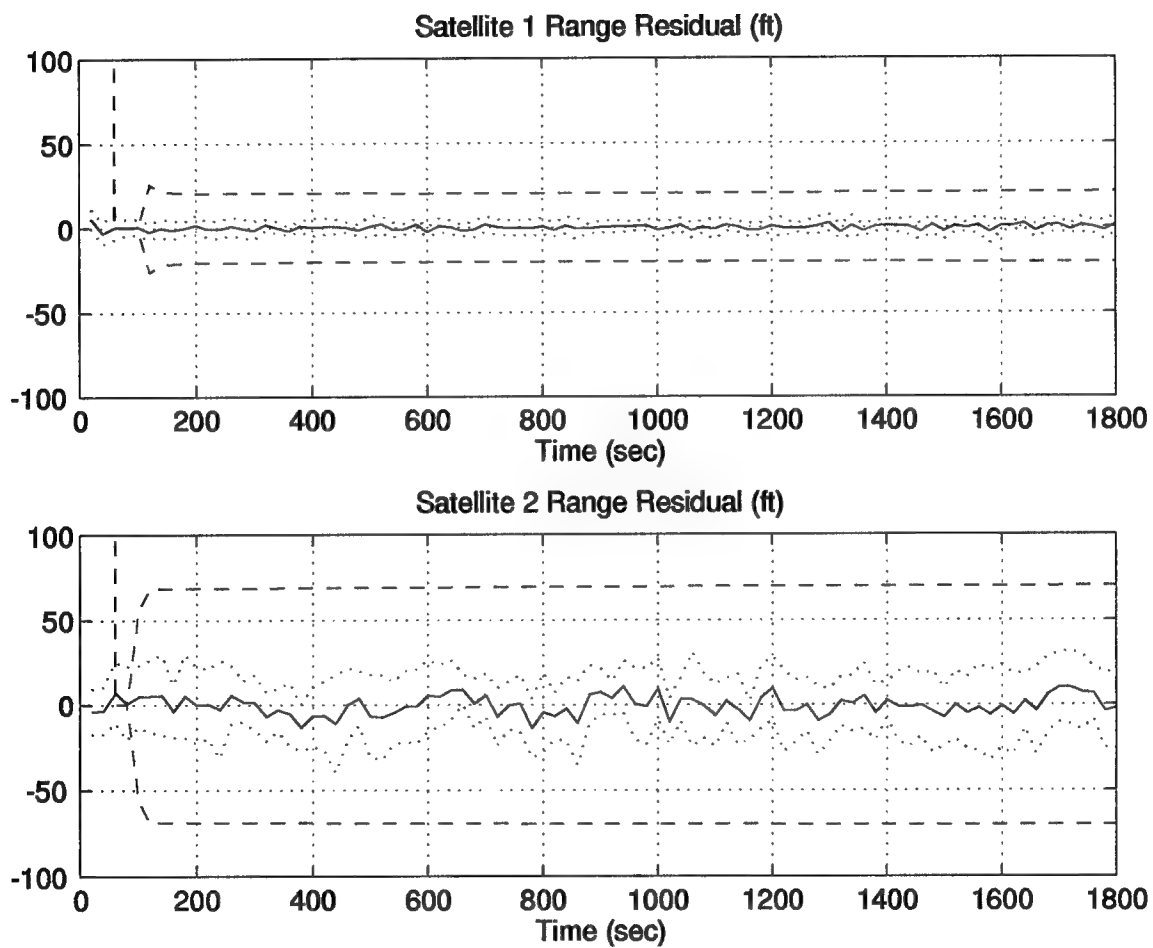
Figure D.6   Residual Errors for the 69-State Filter Computer Simulation
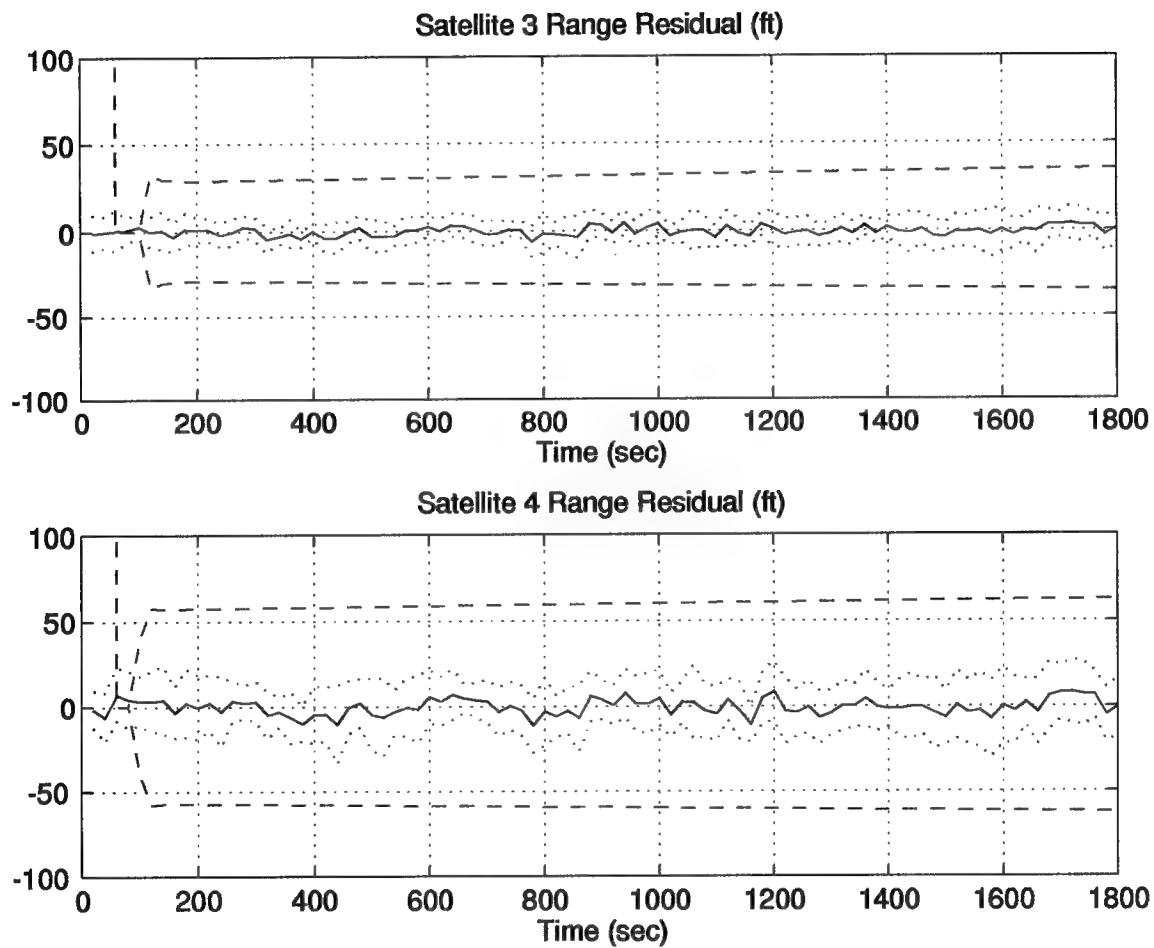
Figure D.7 Residual Errors for the 69-State Filter Computer Simulation

Appendix E. The 13-State Filter Category I Computer Simulation Results

The following plots represent the results of a 10-run Monte Carlo analysis using a 13-state filter.

Table E.1 Legend of Filter Tuning Plots

Symbol	Definition
Solid Line	Mean Error
Dotted Line	Mean Error ± True Sigma
Dashed Line	± Filter Predicted Sigma
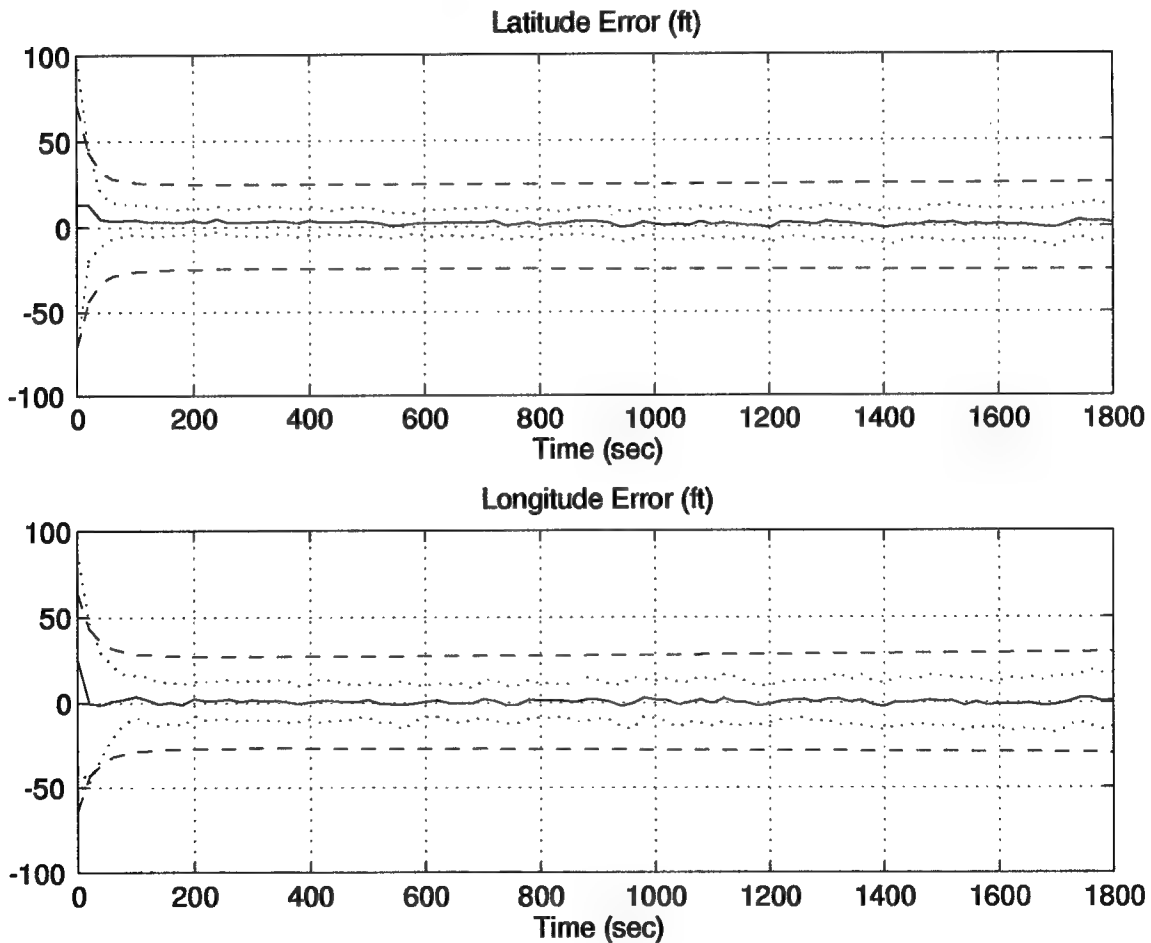
Figure E.1 Position Errors for the 13-State Filter Computer Simulation
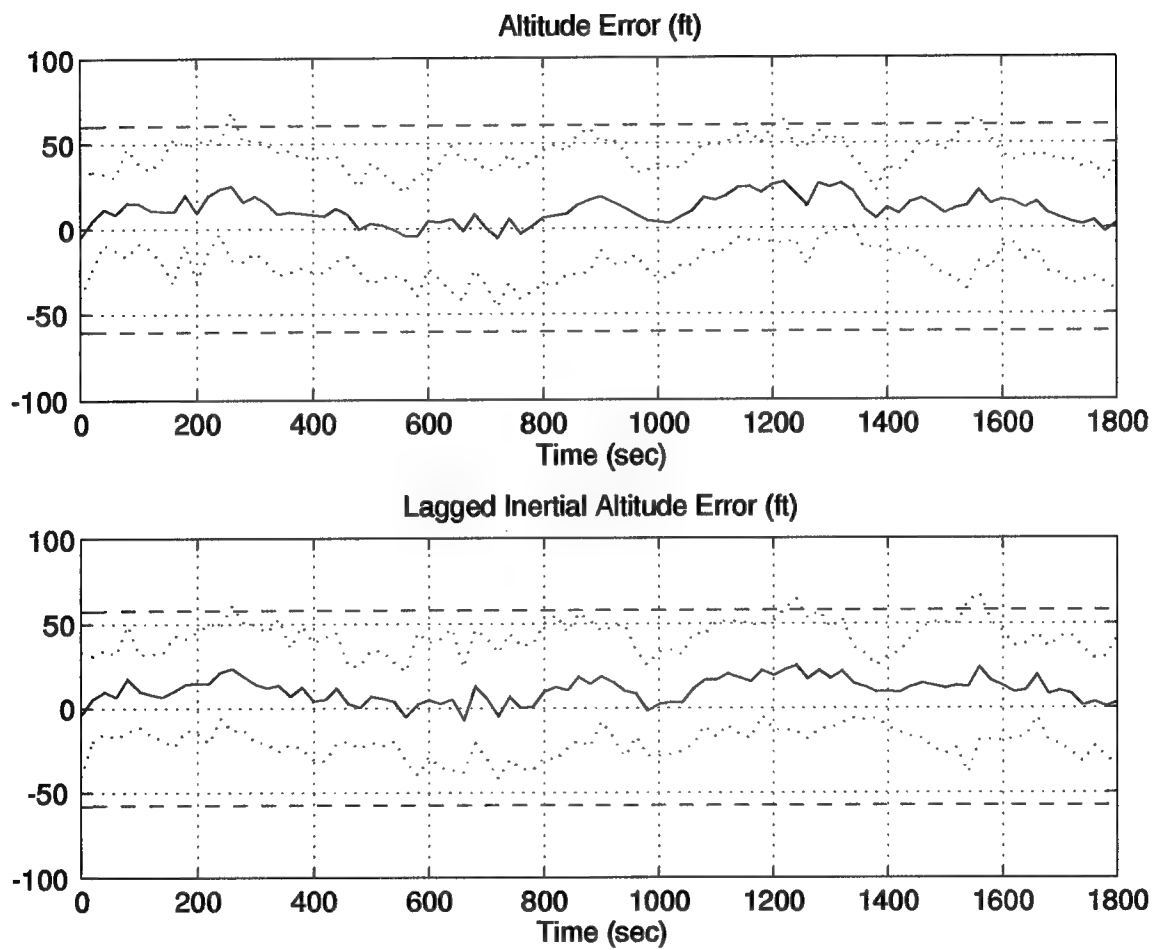
Figure E.2   Altitude Errors for the 13-State Filter Computer Simulation
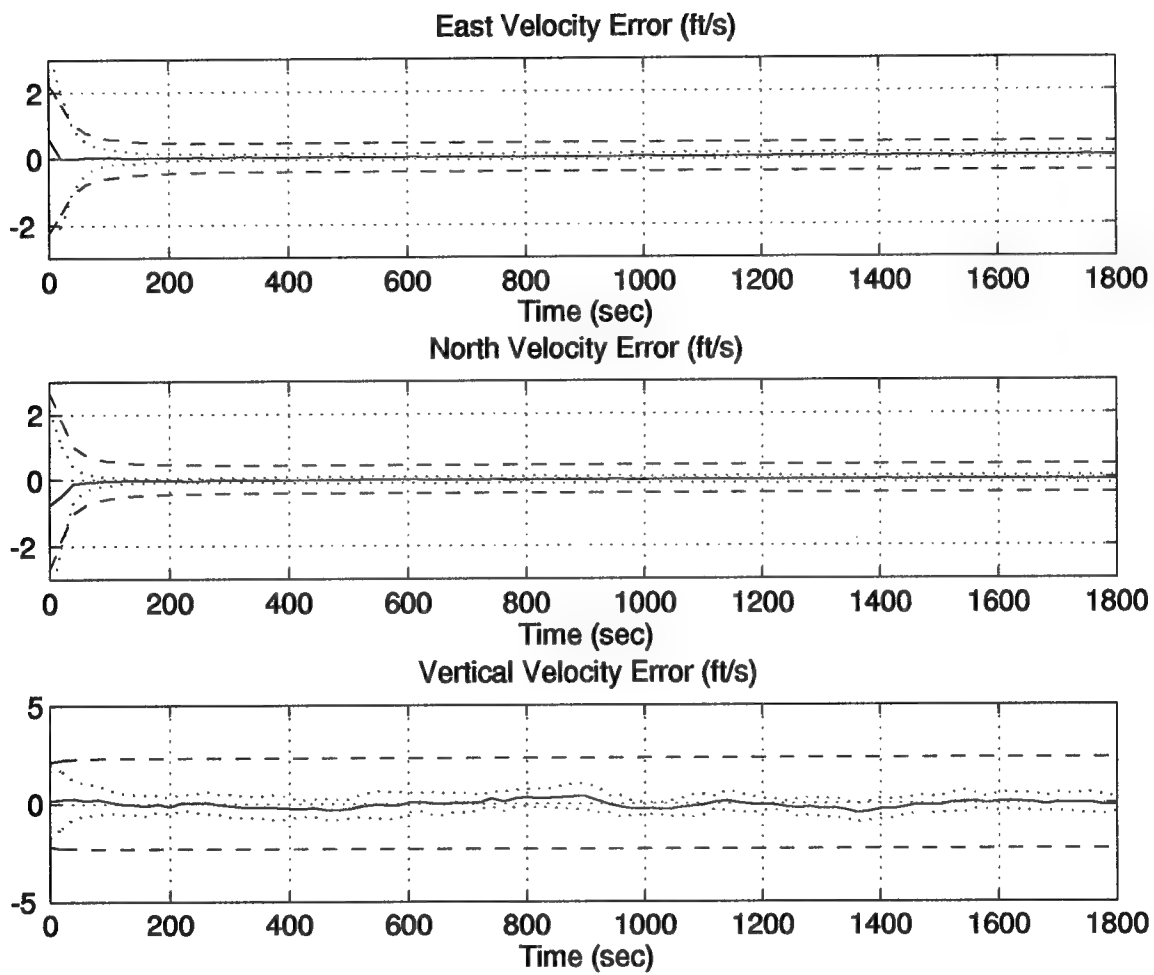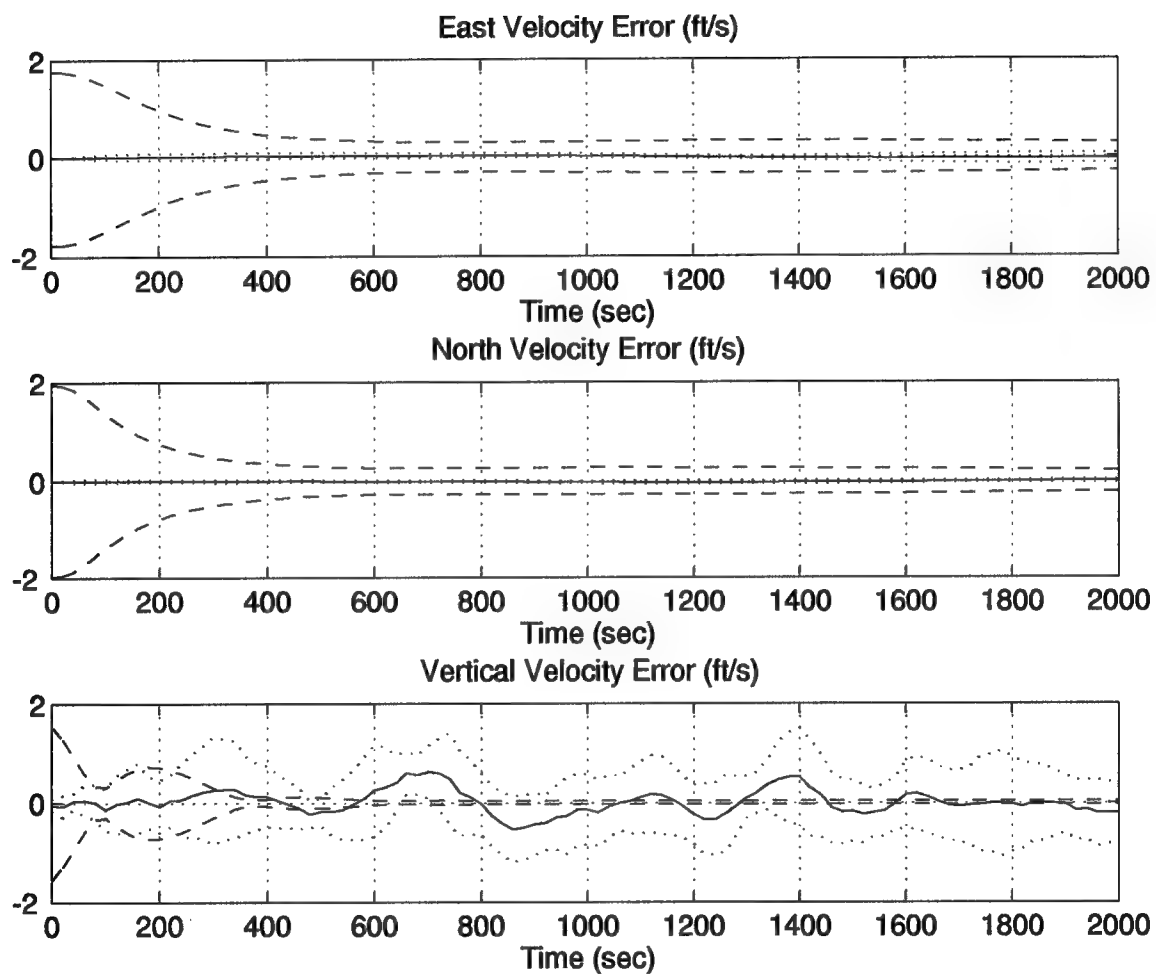
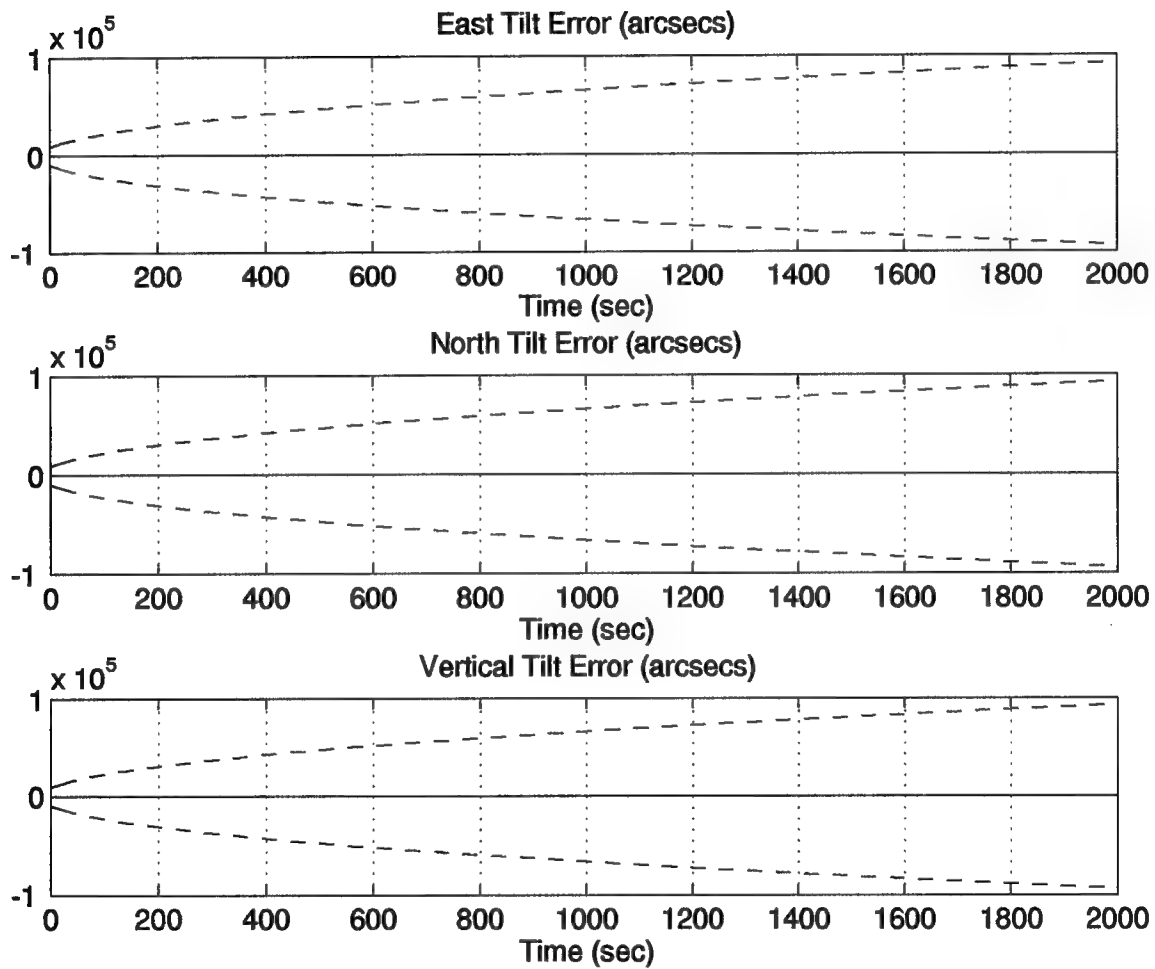Figure E.3   Velocity Errors for the 13-State Filter Computer Simulation
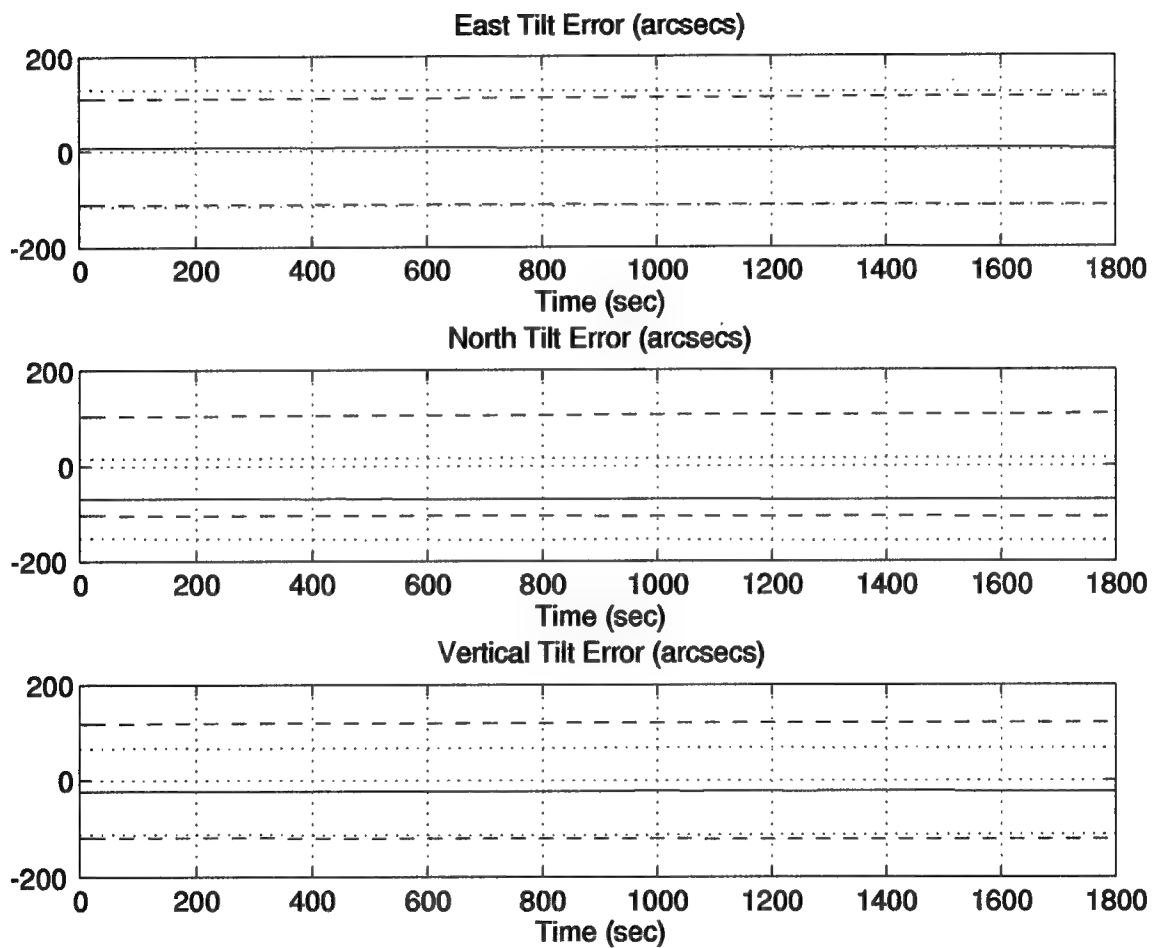
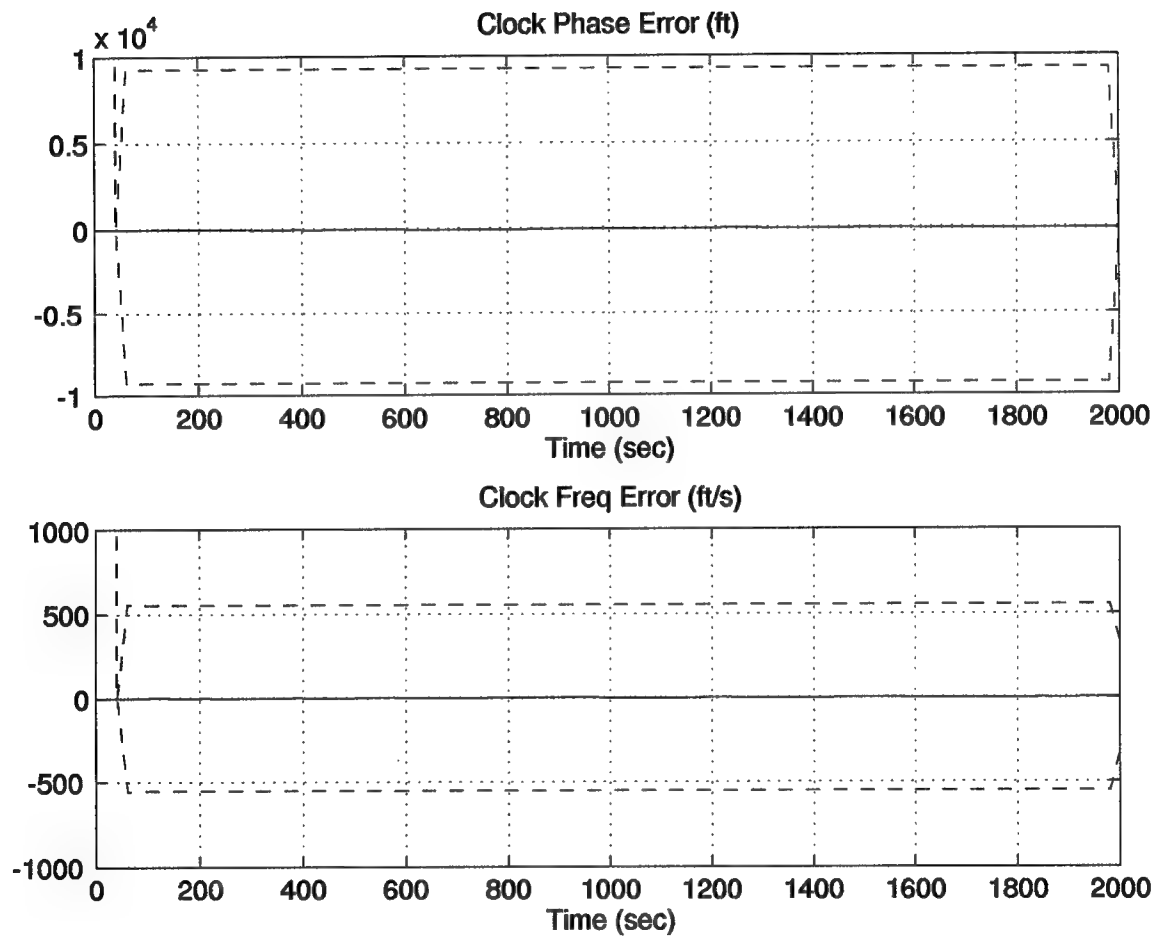Figure E.4   Tilt Errors for the 13-State Filter Computer Simulation
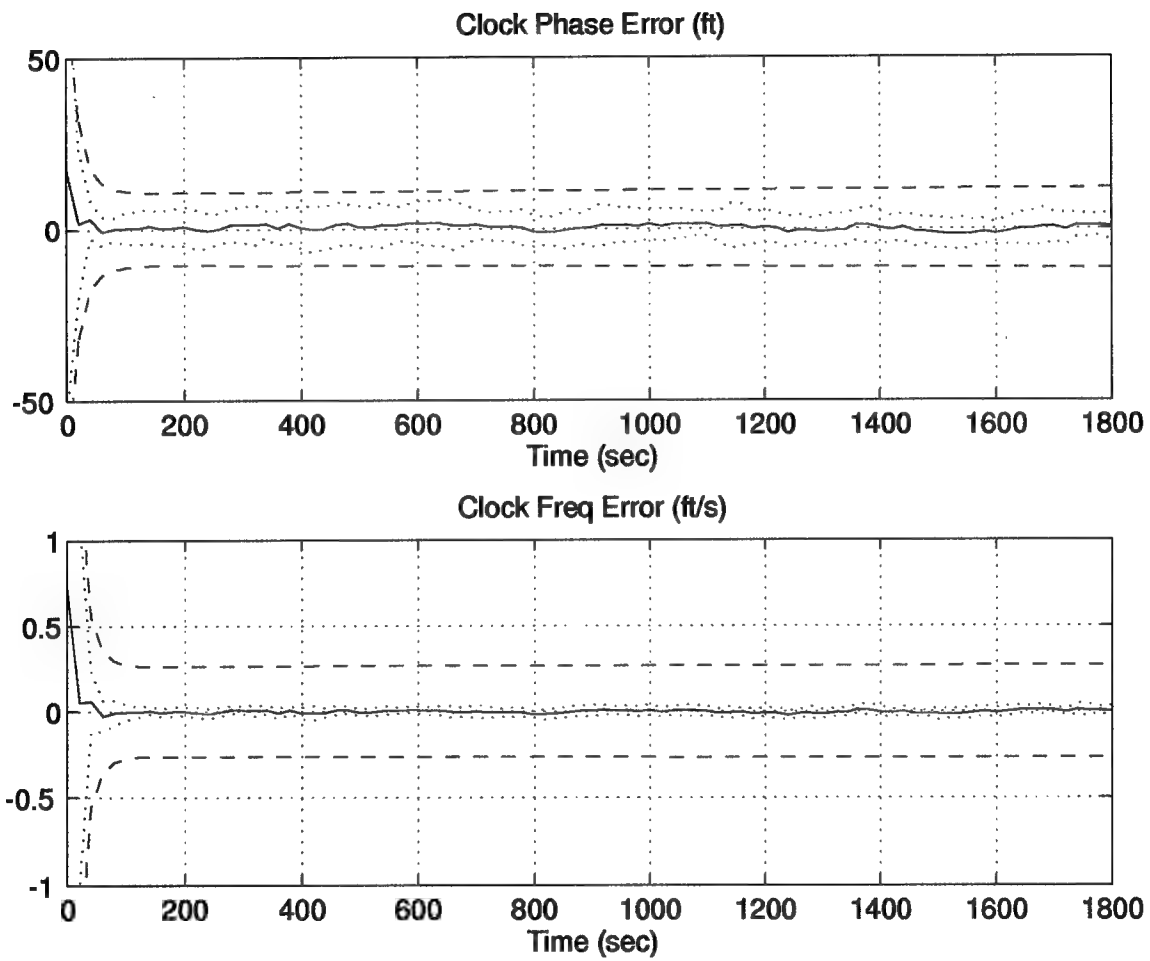
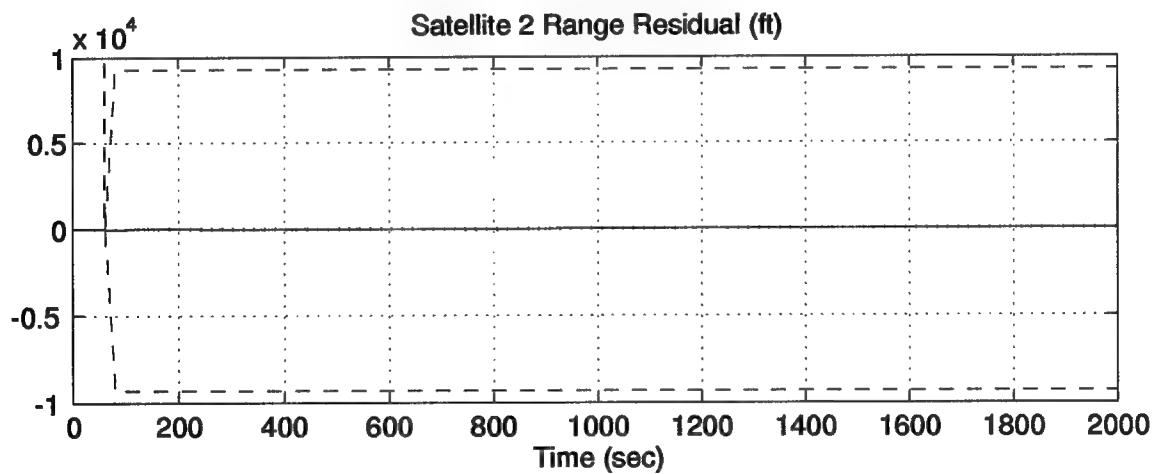Figure E.5   Clock Errors for the 13-State Filter Computer Simulation
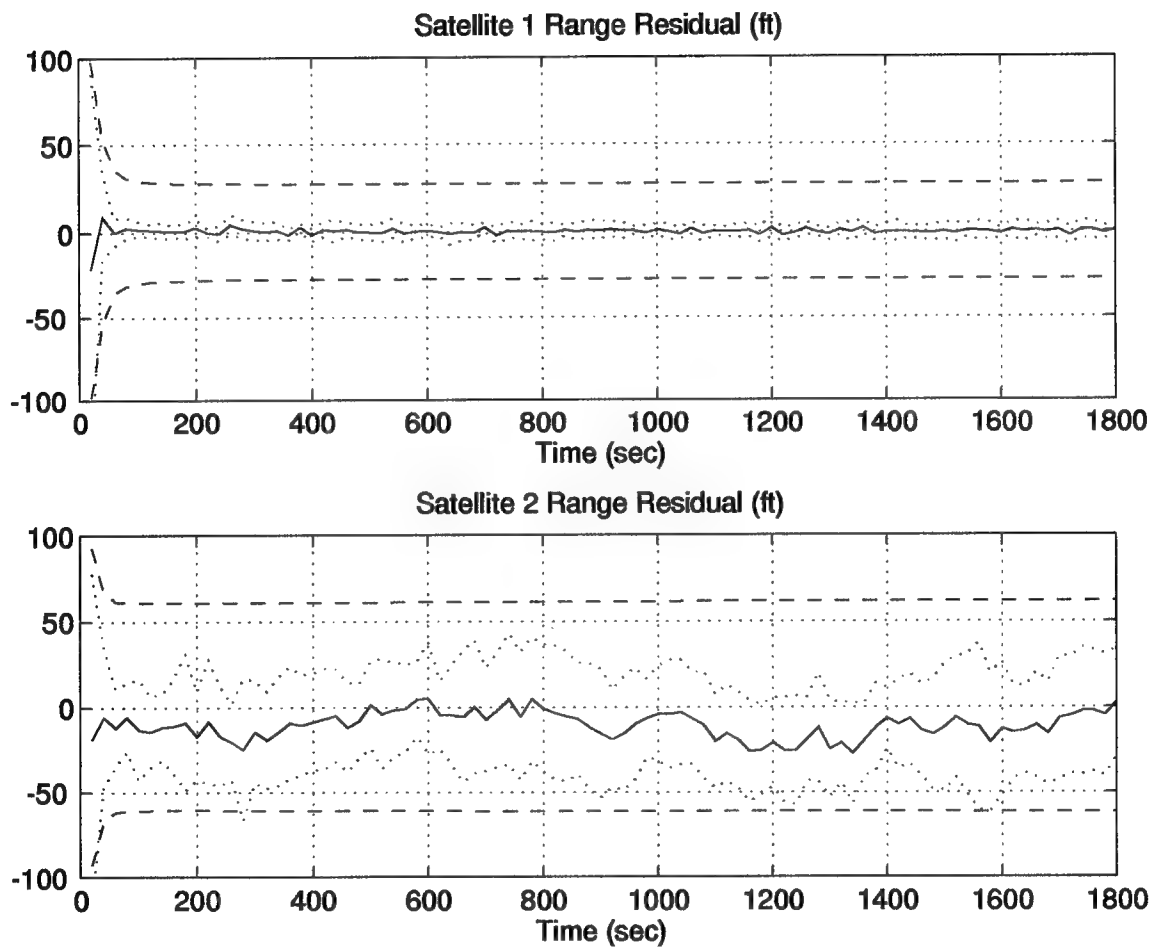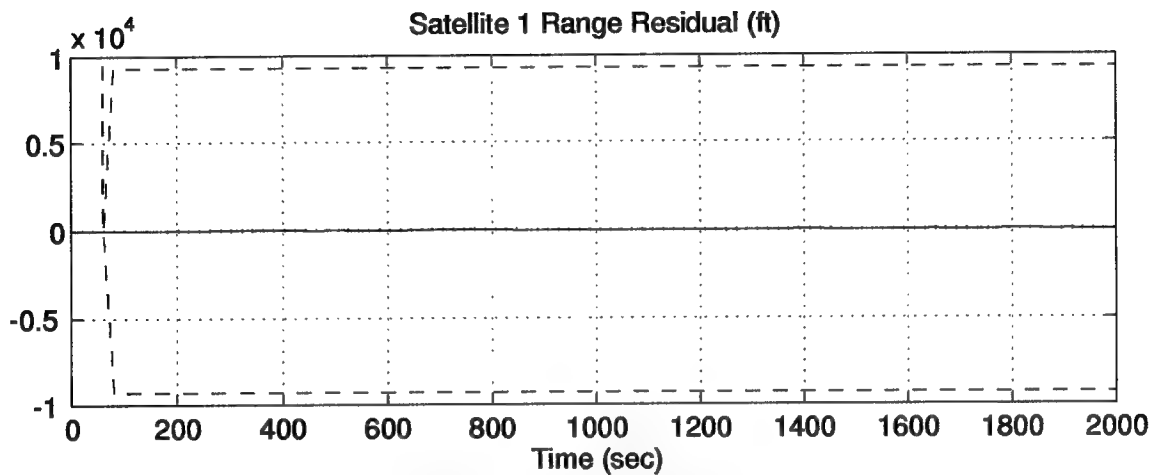
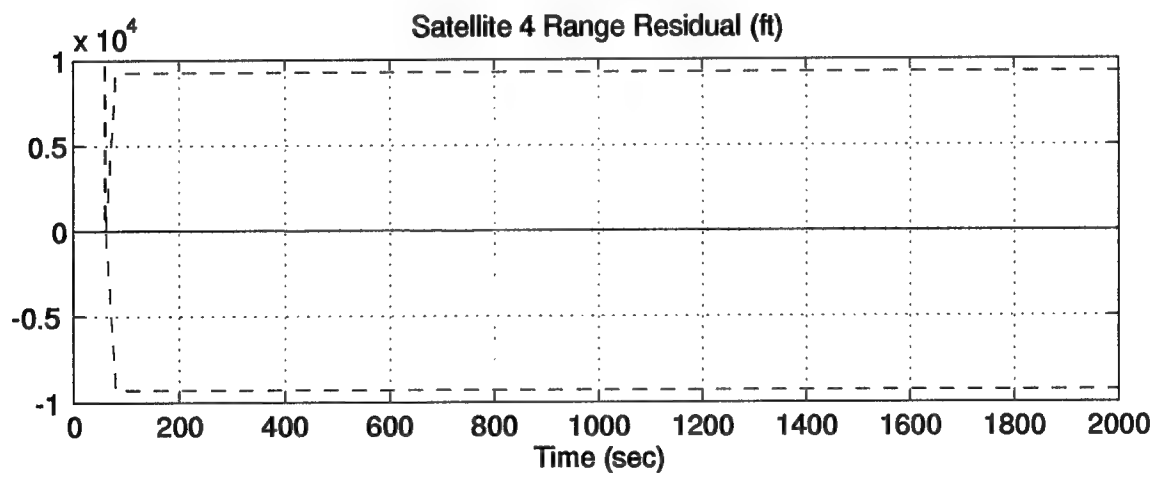Figure E.6   Residual Errors for the 13-State Filter Computer Simulation
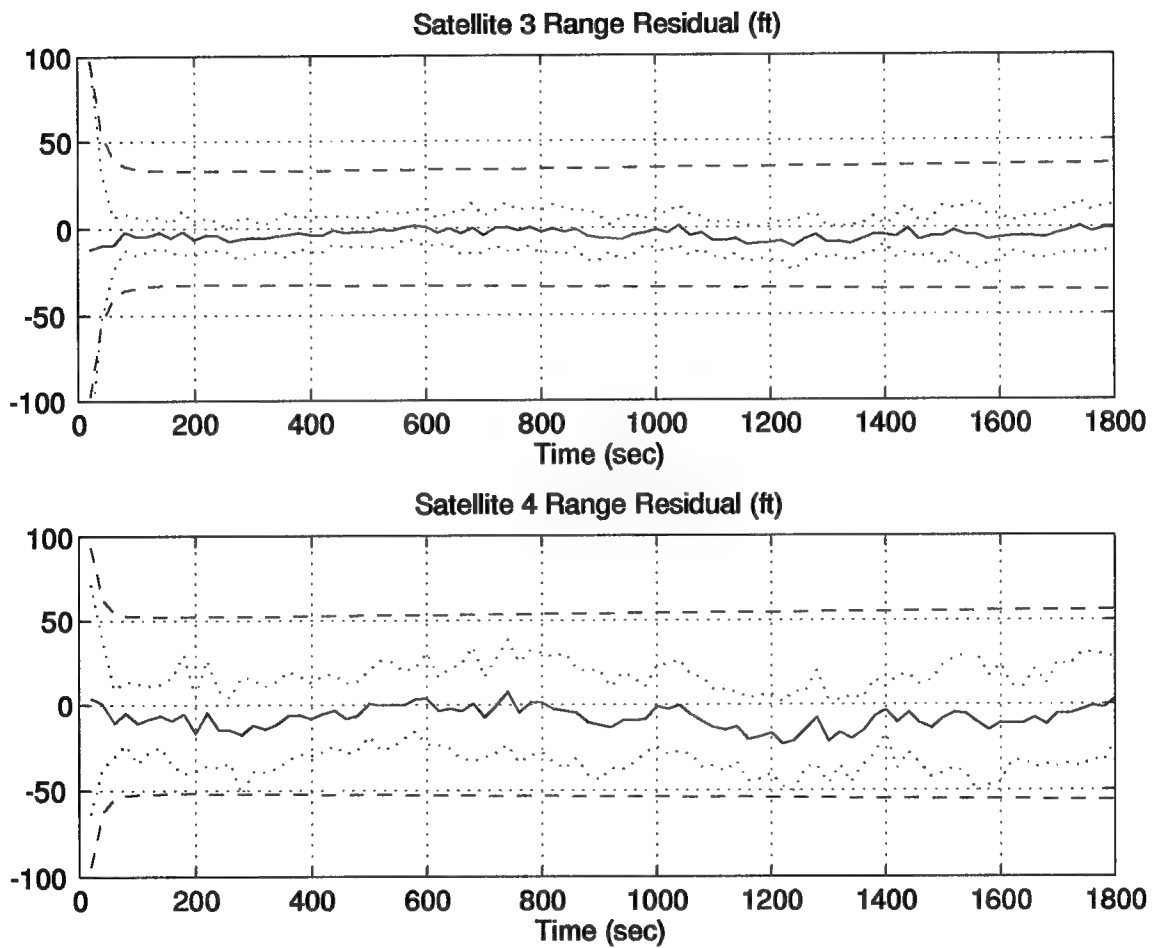
Figure E.7   Residual Errors for the 13-State Filter Computer Simulation

Appendix F. The 69-State Filter Category II Computer Simulation Results

The following plots represent the results of a 10-run Monte Carlo analysis using a 69-state filter. The dynamic driving noise strengths were taken from the values used to tune the filter in the GPS/INS hardware integration.

Table F.1   Legend of Filter Tuning Plots

Symbol	Definition
Solid Line	Mean Error
Dotted Line	Mean Error ± True Sigma
Dashed Line	± Filter Predicted Sigma
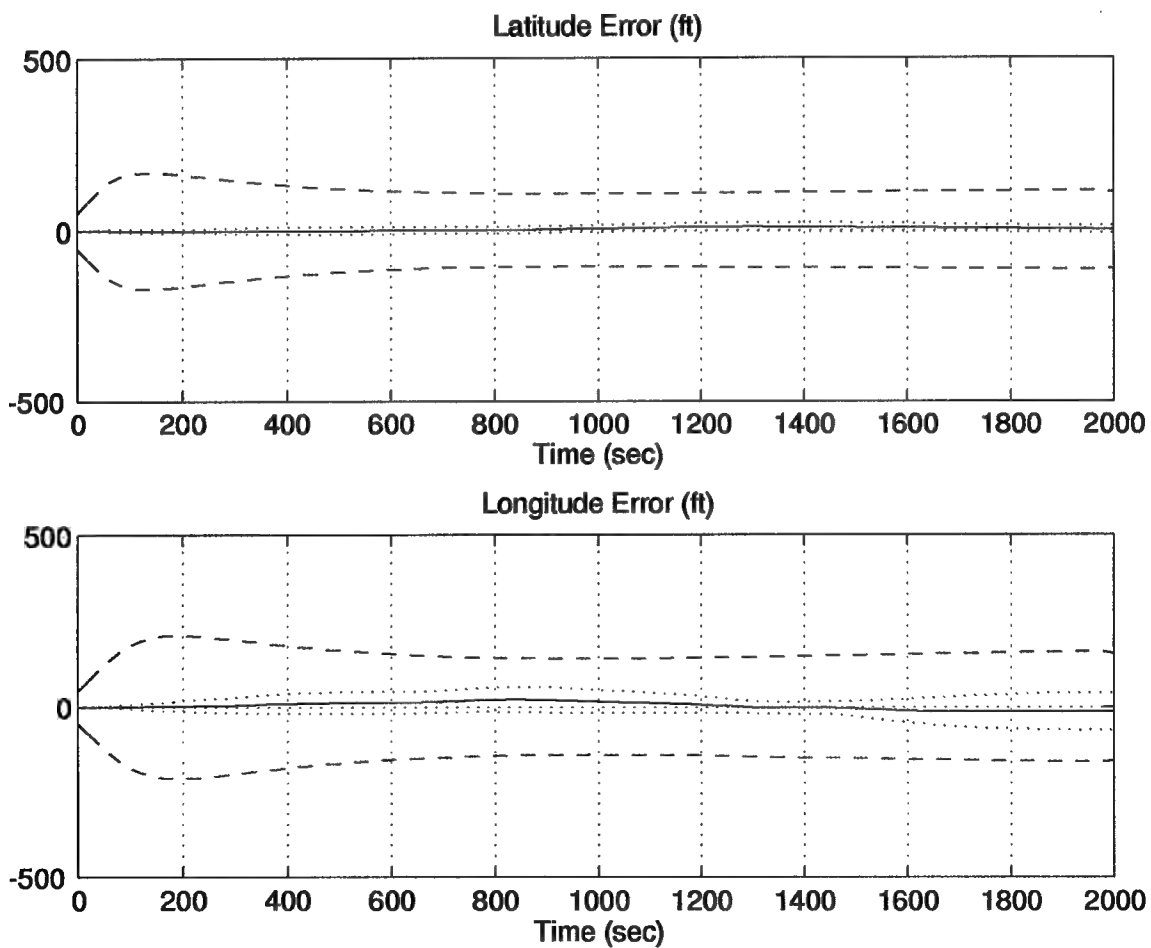
Figure F.1   Position Errors for the 69-State Filter Computer Simulation

Figure F.2   Altitude Errors for the 69-State Filter Computer Simulation

Figure F.3   Velocity Errors for the 69-State Filter Computer Simulation

Figure F.4   Tilt Errors for the 69-State Filter Computer Simulation

F-4

Figure F.5   Clock Errors for the 69-State Filter Computer Simulation

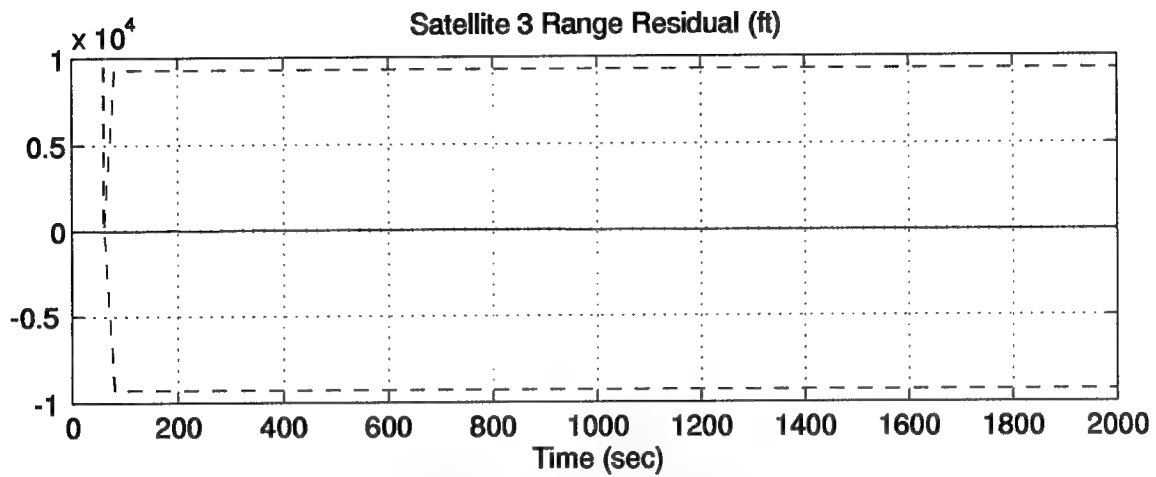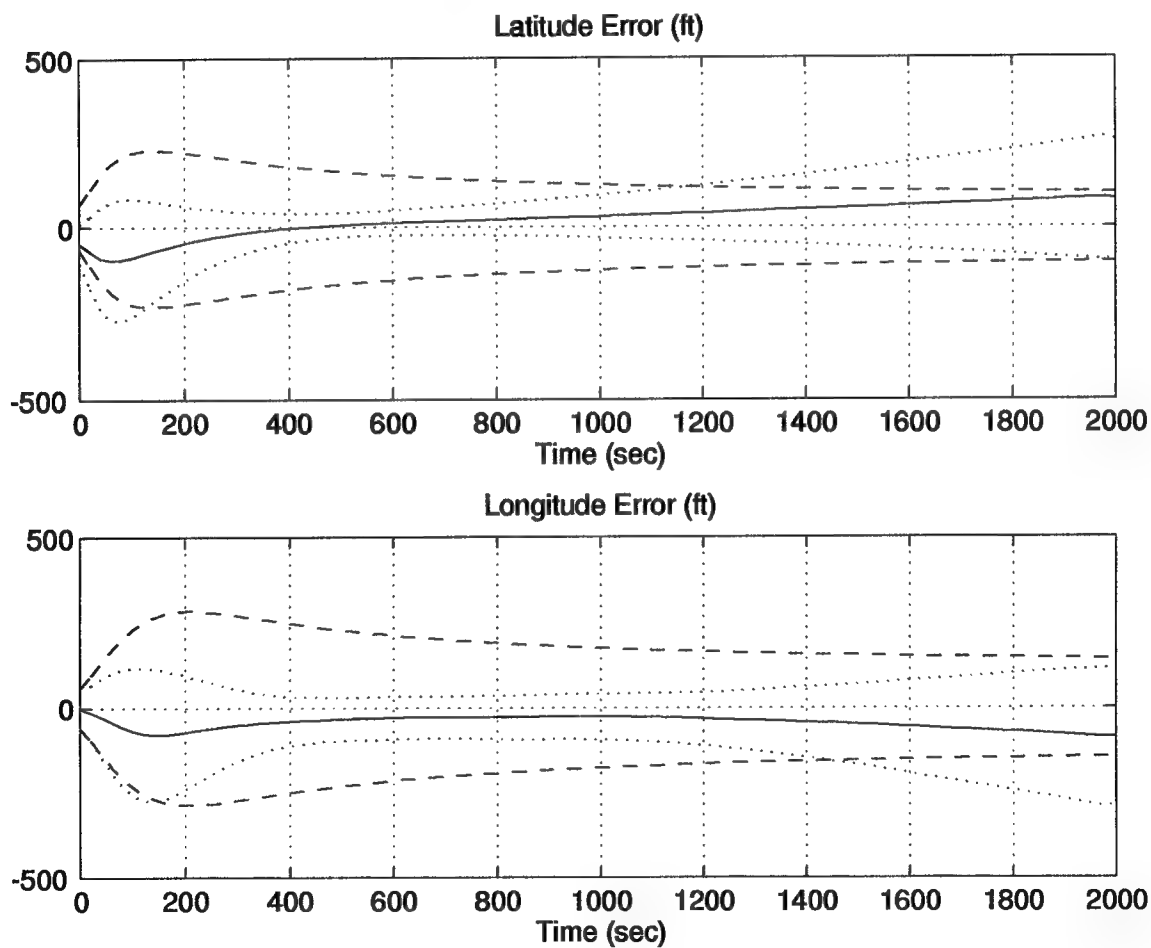Figure F.6 Residual Errors for the 69-State Filter Computer Simulation

Figure F.7   Residual Errors for the 69-State Filter Computer Simulation

Appendix G. The 13-State Filter Category II Computer Simulation Results

The following plots represent the results of a 10-run Monte Carlo analysis using a 13-state filter and the dynamic driving noise strength values from the GPS/INS hardware integration.

Table G.1   Legend of Filter Tuning Plots

Symbol	Definition
Solid Line	Mean Error
Dotted Line	Mean Error ± True Sigma
Dashed Line	± Filter Predicted Sigma

Figure G.1   Position Errors for the 13-State Filter Computer Simulation
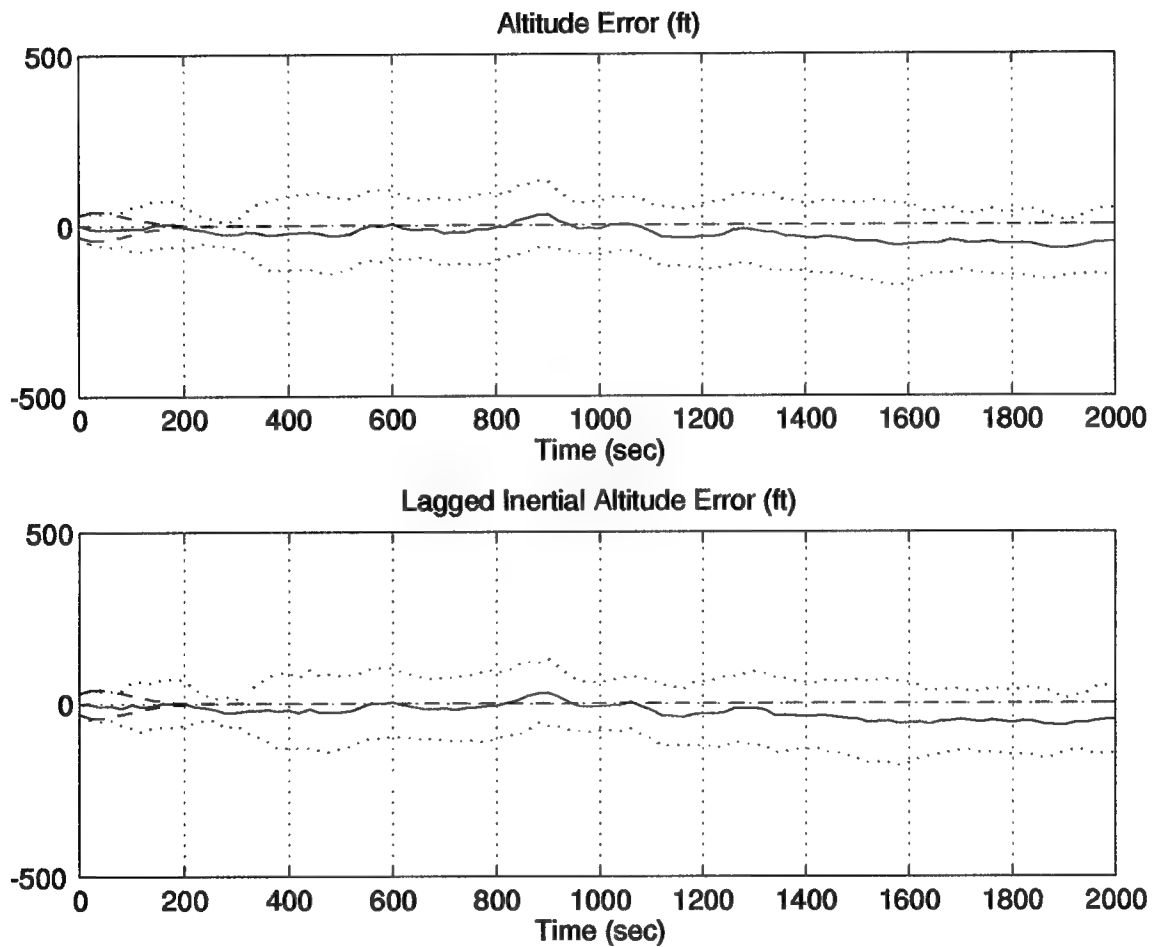
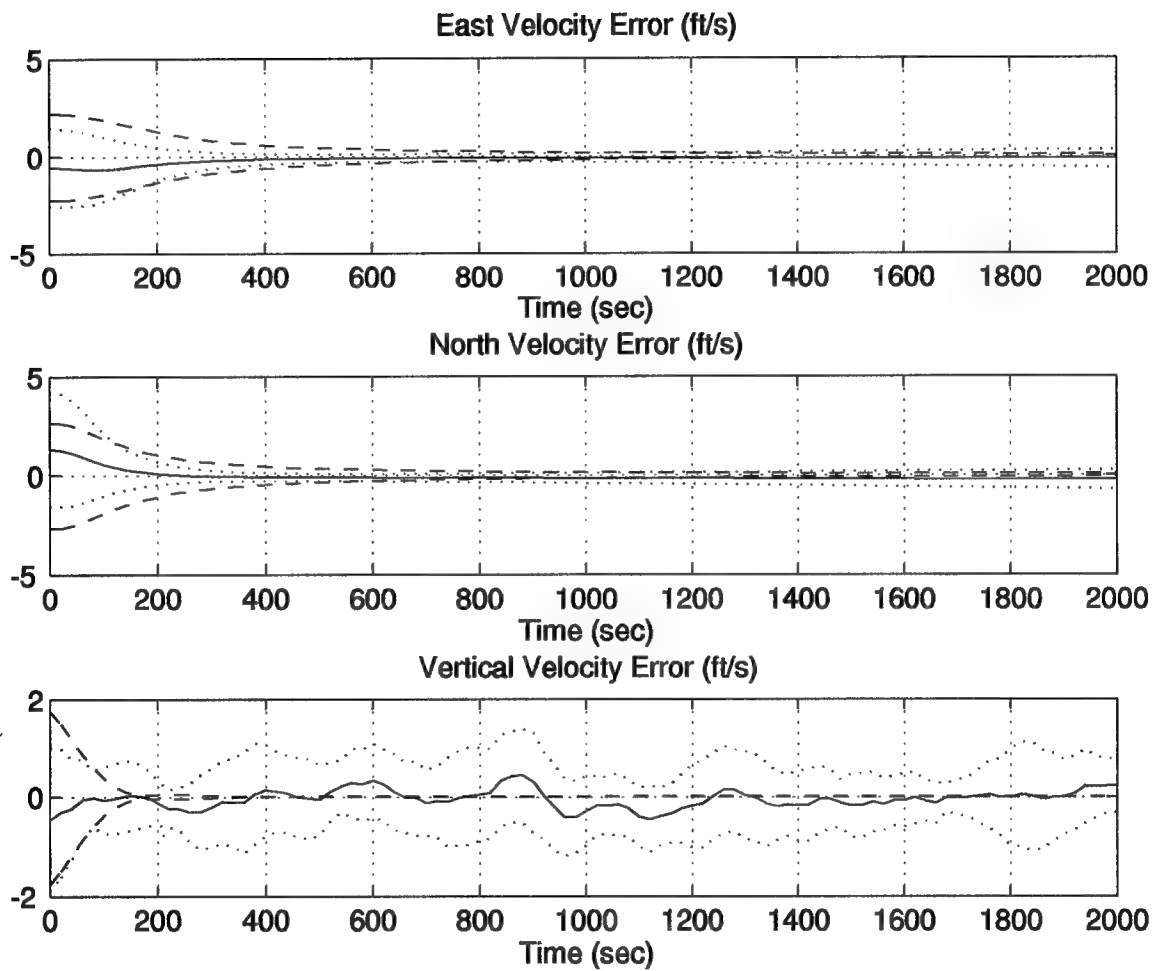Figure G.2   Altitude Errors for the 13-State Filter Computer Simulation

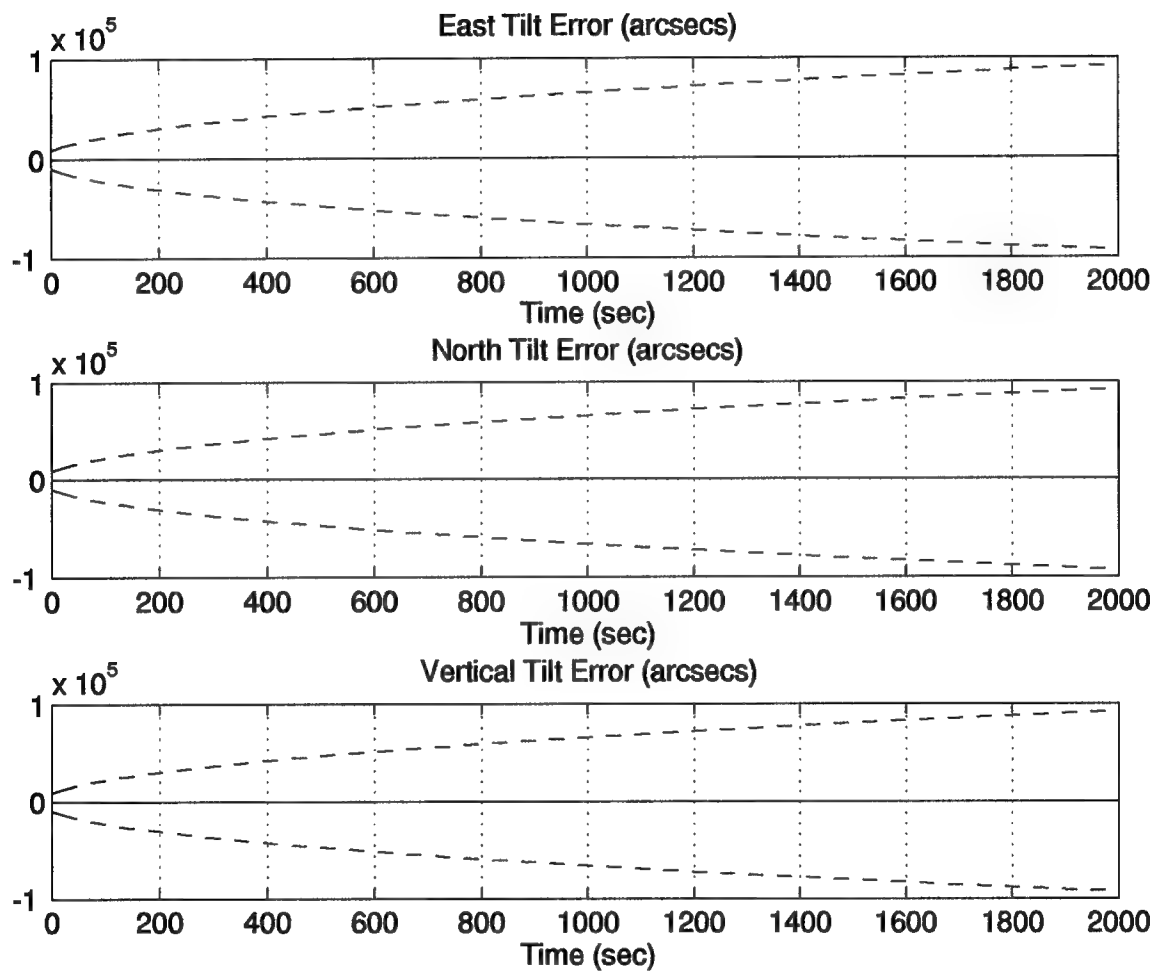Figure G.3   Velocity Errors for the 13-State Filter Computer Simulation

Figure G.4 Tilt Errors for the 13-State Filter Computer Simulation
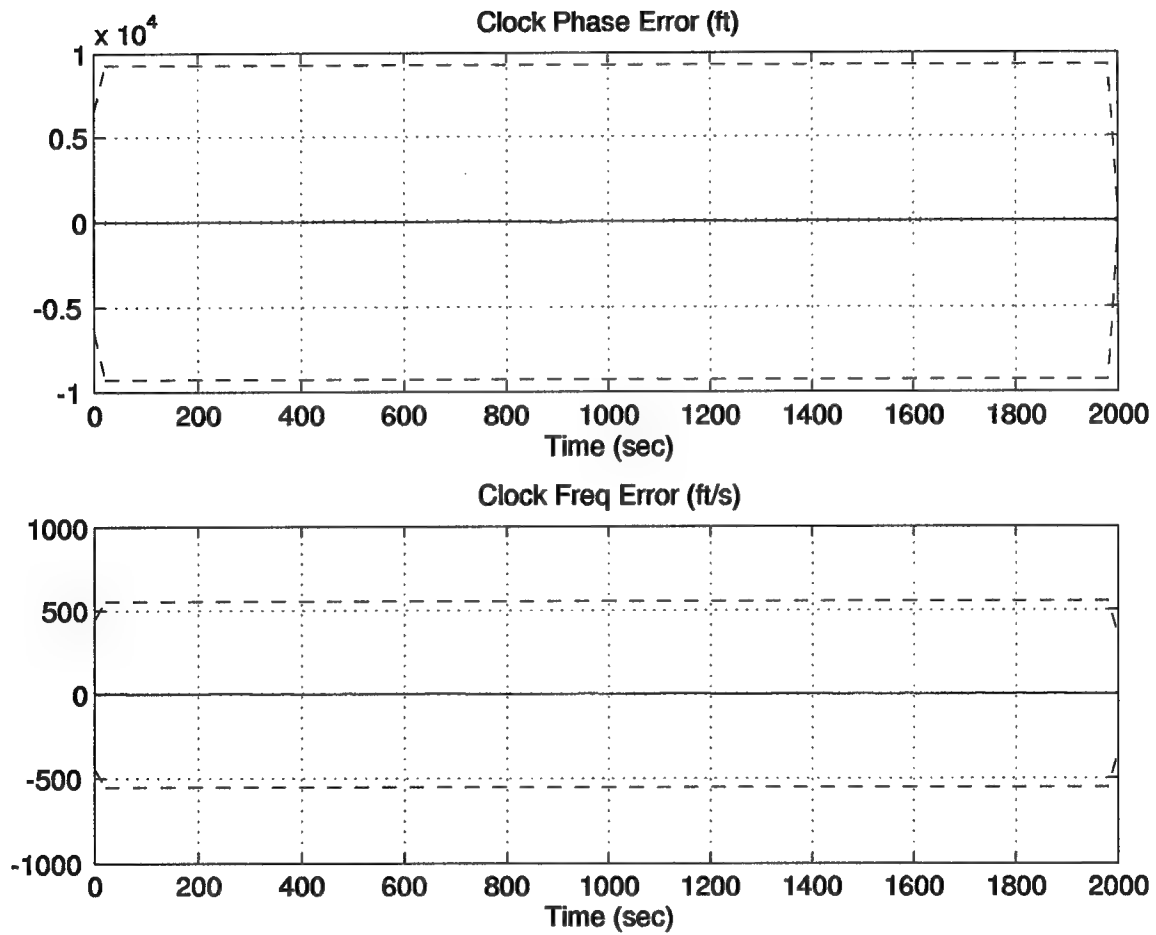
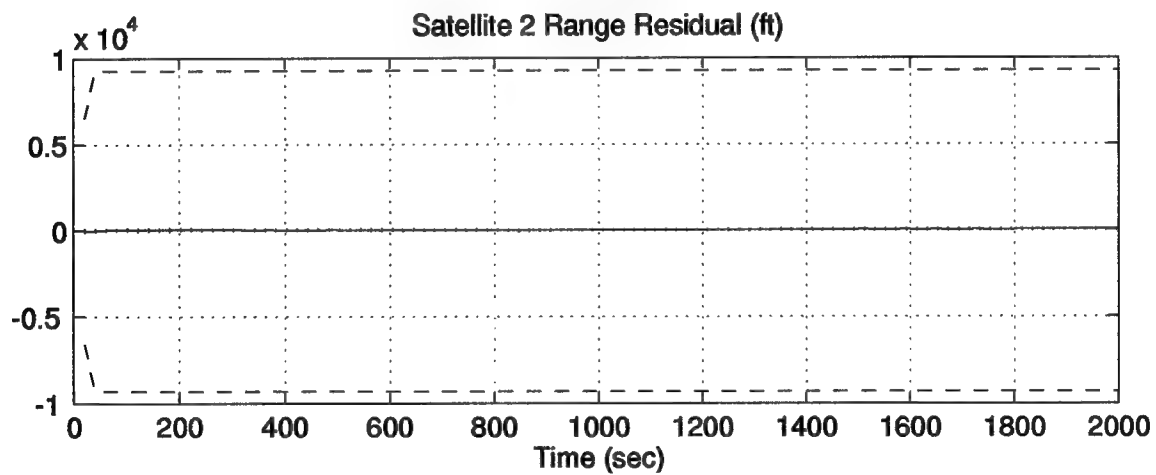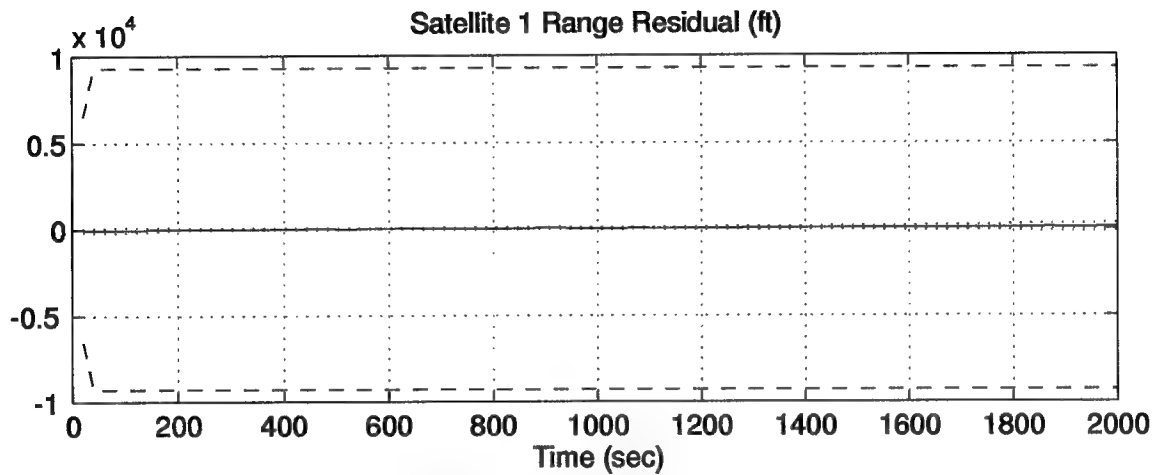Figure G.5   Clock Errors for the 13-State Filter Computer Simulation

Figure G.6   Residual Errors for the 13-State Filter Computer Simulation
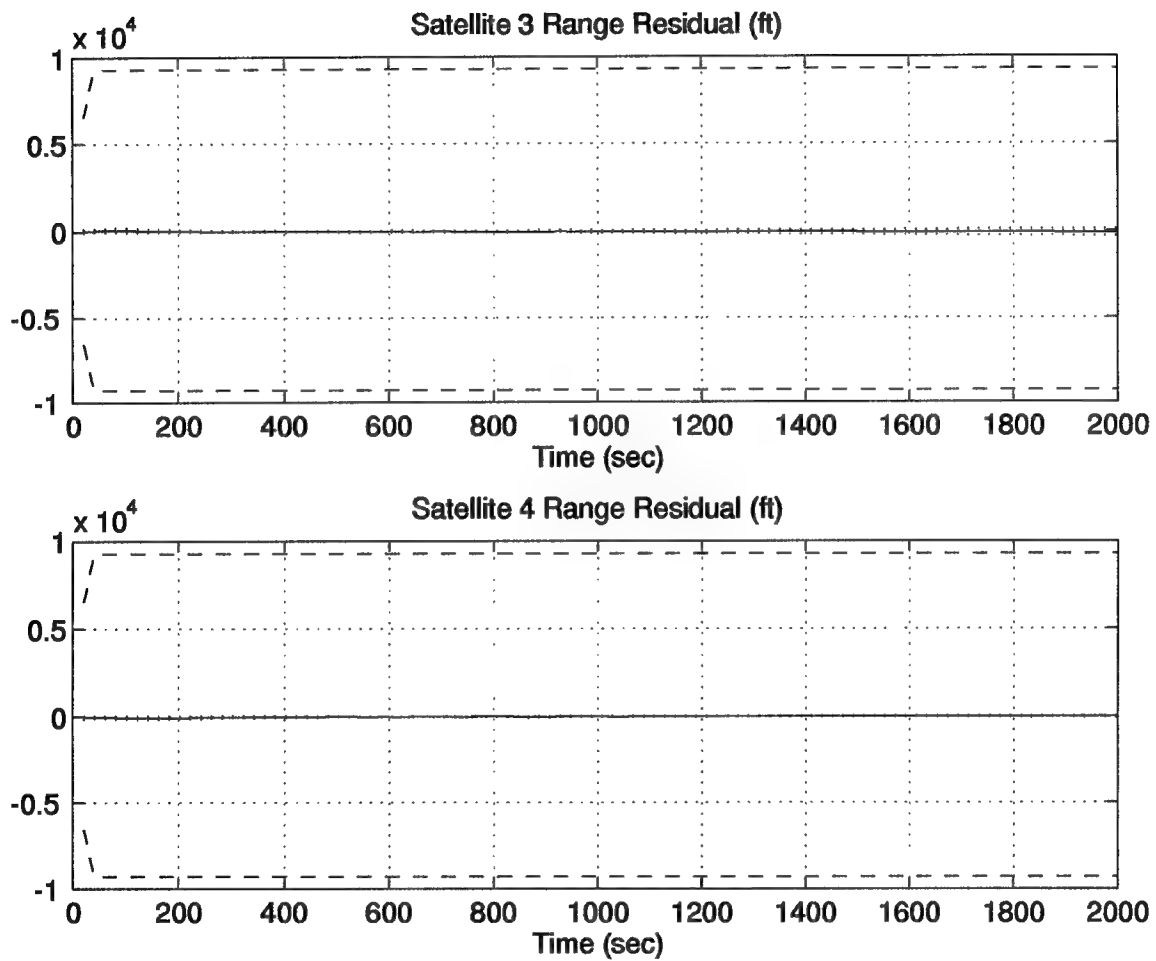
Figure G.7   Residual Errors for the 13-State Filter Computer Simulation

Appendix H. INS Plots

The following plots represent a three hour data set collected from the 1553 data bus for the LN-94. As discussed in Chapter 4, the LN-94 has not had any calibration, alignment or preventive maintenance performed on it in several years; therefore, the large drift rate was expected. This data was collected over a three hour period and was used as input for the GPS/INS integration research.

Figure H.1   LN-94 INS Position Errors

Figure H.2   LN-94 INS Altitude Errors
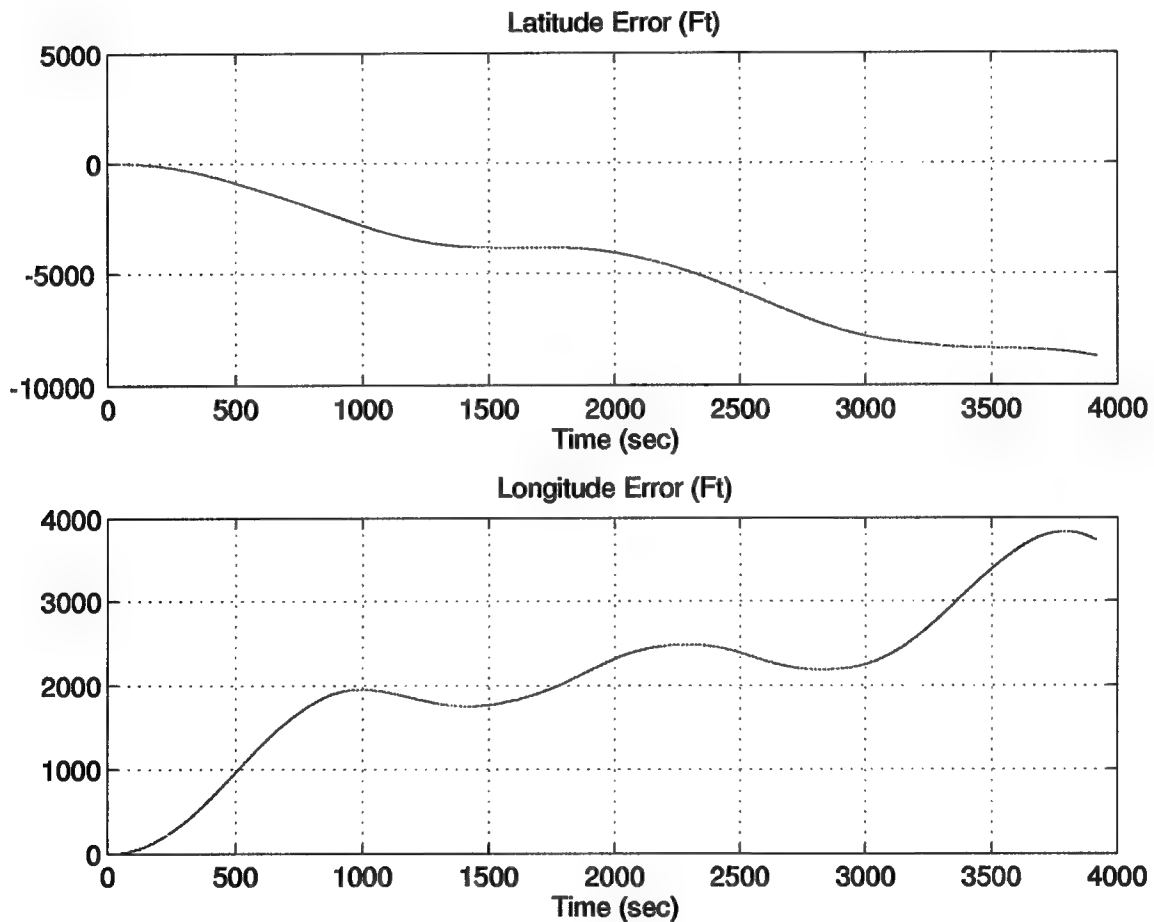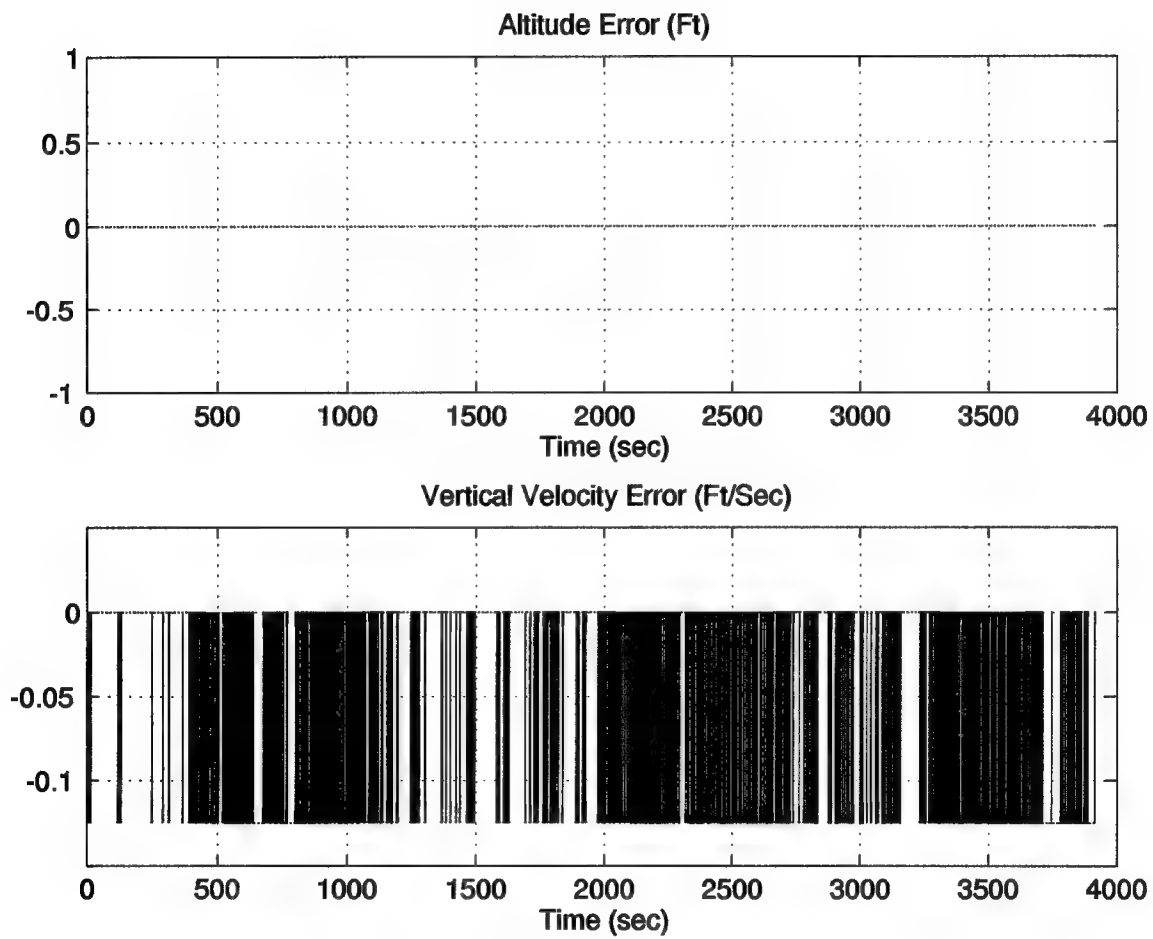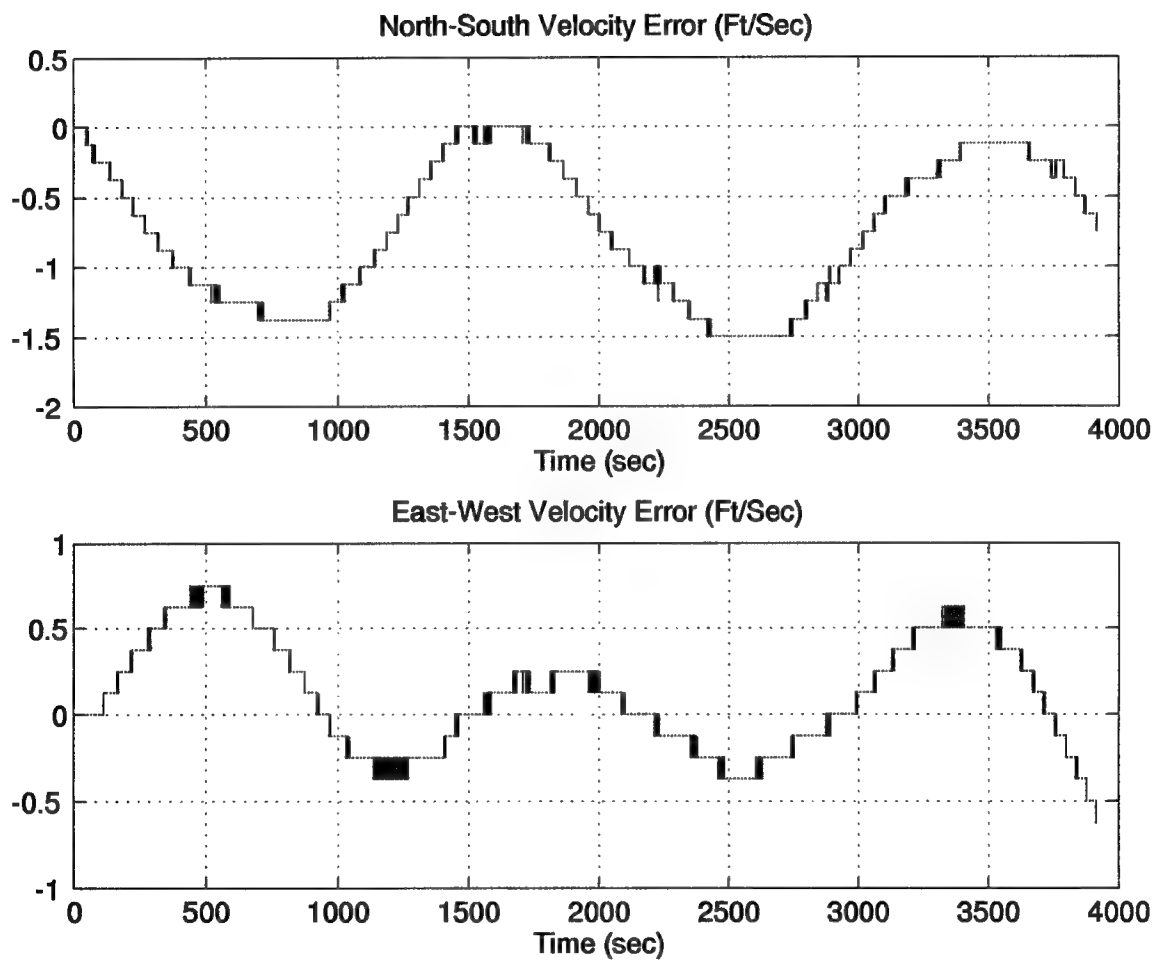
Figure H.3   INS Velocity Errors

Appendix I. Residuals

I.1   Overview

Section 4.4.3 described the sudden bias shifts which occured in the GPS receivers during data collection. The research could not determine if this was a problem with the receiver clocks or an intentional correction by the GPS receiver to maintain some relation to the satellite time. The bias shifts did not appear to be on any specific interval, but rather random in nature. In any event, the phenomena must be accounted for when using actual hardware in a GPS/INS integration. This appendix describes the bias shifts which occured and shows the results.

Recall from Section 1.2.3, the primary component of error in the actual range from the satellite to the user is the difference between the GPS and receiver times. A very slight difference in time can create a very huge error, since the difference is multiplied by the speed of light to provide a distance error. During data collection, the GPS receivers occasionally induced a bias shift at random intervals. Below is an excerpt from a data file showing the bias shift. Each column represents pseudoranges from each of seven satellites. The starred rows indicate where the clock bias shifted.

Satellite ID #

19	22	27	29	18	28	31
1.0e+07 *						
5.4049	5.9398	6.2381	5.2932	5.0767	5.1417	5.0822
5.3977	5.9392	6.2296	5.2920	5.0734	5.1379	5.0755
5.3904	5.9385	6.2212	5.2908	5.0702	5.1341	5.0688
5.3831	5.9379	6.2127	5.2896	5.0669	5.1303	5.0621
5.3759	5.9373	6.2043	5.2884	5.0637	5.1265	5.0554
5.3686	5.9366	6.1958	5.2872	5.0605	5.1227	5.0488
5.3614	5.9360	6.1874	5.2861	5.0573	5.1190	5.0421
5.3541	5.9353	6.1789	5.2849	5.0541	5.1152	5.0355
5.3469	5.9347	6.1705	5.2838	5.0510	5.1115	5.0289

5.3396	5.9340	6.1620	5.2827	5.0479	5.1078	5.0223
5.3324	5.9334	6.1536	5.2815	5.0447	5.1041	5.0157
5.3252	5.9327	6.1451	5.2804	5.0416	5.1005	5.0091
5.3180	5.9321	6.1367	5.2793	5.0385	5.0968	5.0026
5.3108	5.9314	6.1282	5.2783	5.0354	5.0932	4.9960
5.3035	5.9308	6.1197	5.2772	5.0324	5.0896	4.9895
5.2963	5.9301	6.1113	5.2761	5.0293	5.0860	4.9830
5.2891	5.9295	6.1028	5.2751	5.0263	5.0824	4.9765
5.2819	5.9288	6.0944	5.2740	5.0233	5.0788	4.9700
5.2747	5.9282	6.0859	5.2730	5.0203	5.0753	4.9635
5.2675	5.9276	6.0775	5.2720	5.0173	5.0717	4.9571
5.2603	5.9269	6.0690	5.2710	5.0143	5.0682	4.9506
5.2532	5.9263	6.0606	5.2700	5.0114	5.0647	4.9442
5.2460	5.9256	6.0521	5.2690	5.0084	5.0612	4.9378
5.2388	5.9250	6.0436	5.2680	5.0055	5.0577	4.9314
*5.2317	5.9243	6.0352	5.2670	5.0026	5.0542	4.9250
*7.0803	7.7795	7.8825	7.1219	6.8555	6.9066	6.7745
7.0732	7.7789	7.8741	7.1209	6.8527	6.9032	6.7681
7.0660	7.7782	7.8656	7.1200	6.8498	6.8997	6.7618
7.0589	7.7776	7.8571	7.1190	6.8470	6.8963	6.7555
7.0517	7.7769	7.8487	7.1181	6.8441	6.8929	6.7492
7.0446	7.7763	7.8402	7.1172	6.8413	6.8896	6.7429
7.0375	7.7756	7.8318	7.1163	6.8385	6.8862	6.7366
7.0303	7.7750	7.8233	7.1154	6.8357	6.8829	6.7303
7.0232	7.7743	7.8148	7.1145	6.8330	6.8795	6.7241
7.0161	7.7737	7.8064	7.1137	6.8302	6.8762	6.7178

This pseudorange shift was approximately 18,500,000 feet. The shift could not be detected in the position plots as shown in Figure I.1; however, the residuals clearly indicate the addition of the bias in Figure I.2.
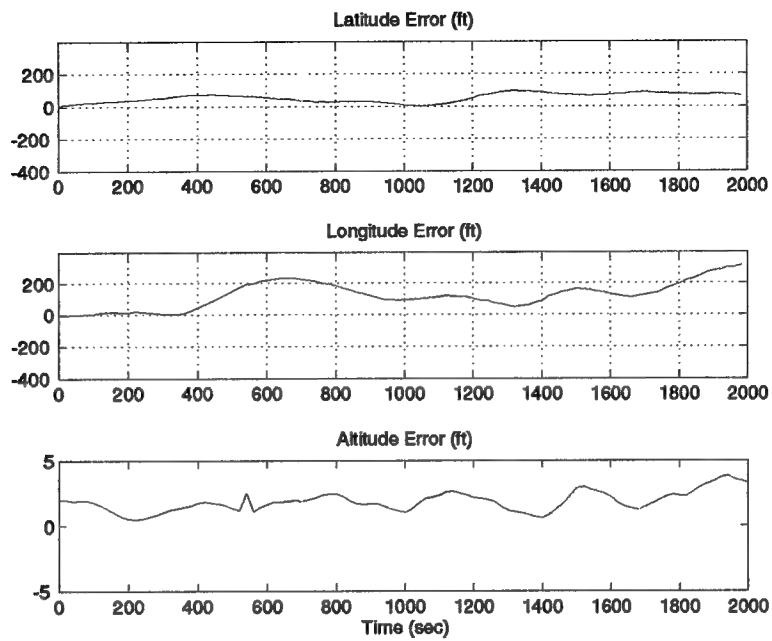
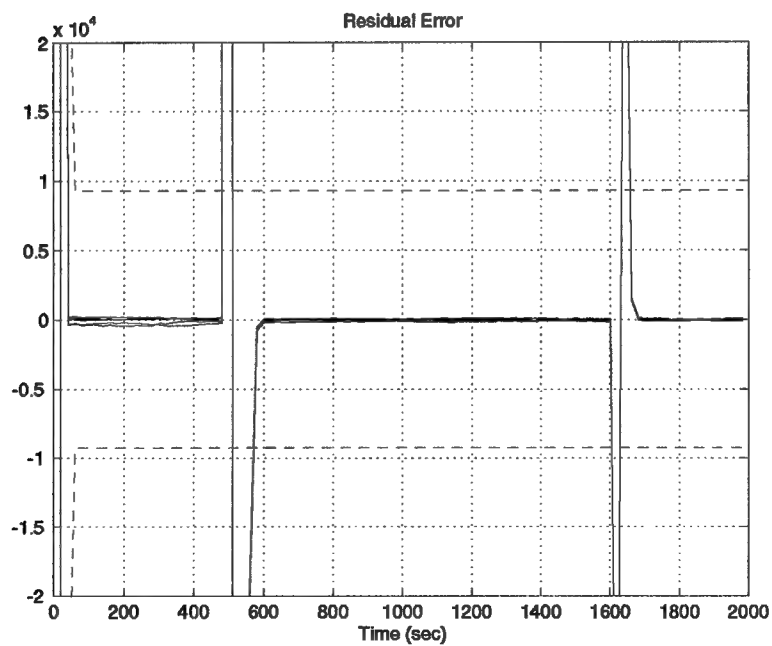Figure I.1   Position Errors Using Data With A GPS Clock Bias Shift

Figure I.2   Residuals Indicating GPS Receiver Clock Bias Shift

I-3

Appendix J. MATLAB Multimode Simulation for Optimal Filter Evaluation (MATSOFE)

J.1 Overview

In an effort to create a more user-friendly and productive simulation tool, the Navigation, Guidance and Control Section at AFIT has developed a collection of m-files known as MATSOFE (7). This Appendix will describe the methodology, m-file structure, differences from MSOFE (3), and the advantages/disadvantages of MATSOFE.

J.2 Background

The name MATSOFE was chosen to reflect the influence MSOFE (3) had upon its development. The MATSOFE tool was created using MSOFE as a basic guideline; wherever possible the Fortran code was converted directly into the MATLAB language. Without the excellent notation in MSOFE, the relatively quick development of MATSOFE would not have been possible. MSOFE has been created as a general tool for solving virtually any type of Kalman filtering problem. Although MATSOFE may eventually grow to encompass all types of problems, to date only the MSOFE manual's orbit and GPS/INS integration problems have been evaluated using the MATLAB tool /citemsofe, matsofe.

J.3 MATSOFE Structure

MATLAB is an interactive system and programming language for general scientific and technical computation /citematlab. MATSOFE has been written as a set of MATLAB scripts and functions using m-files /citematsofe. The m-file is a feature of MATLAB used in the MATSOFE tool, allowing the user to execute sequences of commands that are stored in files with names that have an extension of .m, as in matekf2.m.

MATSOFE is a program, similar to MSOFE, which allows the user to design integrated systems via Kalman filtering techniques and to evaluate their performance. MATSOFE permits the designer to implement his own truth and filter models. The remainder of this section will present the flow of the MATSOFE algorithm and indicate differences from its MSOFE counterpart. The hierarchical structure of MATSOFE is depicted in Figure J.1. This figure gives a depiction of
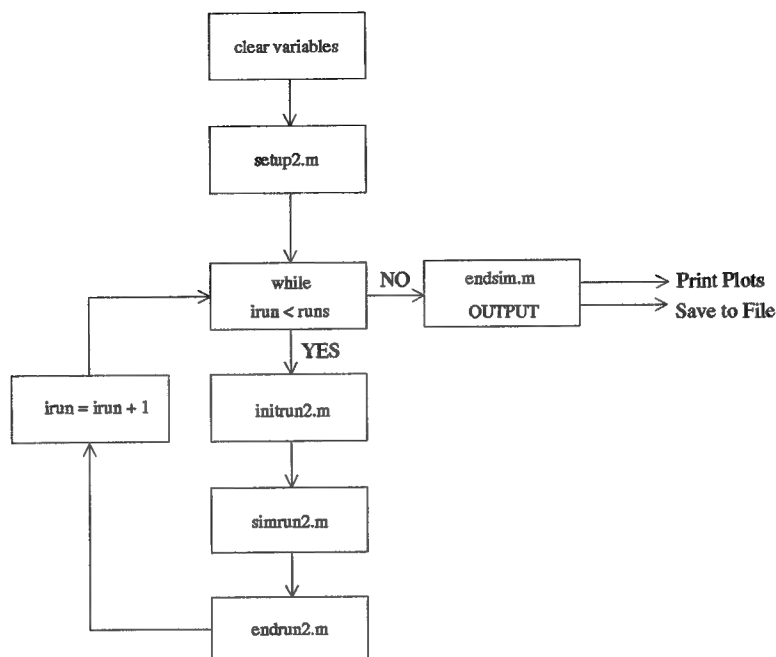
Figure J.1   Block Diagram of MATSOFE Hierarchical Structure

MATSOFE in its most general form with all of its top level subroutines.

J.4   Setting Up MATSOFE

This section describes the setup of MATSOFE for a GPS/INS integration simulation run. In order to setup MATSOFE initially for a new problem, the following m-files must be changed: setup2.m, sizes.m, initcond.m, newprop.m, newmeas.m, ffform2.m, fsform2.m, endrun.m, and endsim.m. These m-files are described in Section J.6. The dynamics models are input into newprop.m, and the measurement equations are entered into newmeas.m. The initial conditions are found in initcond.m, sizes.m, and setup2.m. The remaining m-files contain the basic Kalman filtering equations and in general do not need alteration.

A comment on initial conditions is required at this point. MSOFE uses two different modes in establishing initial conditions (3). Mode 1 assumes that the run starts when the filter is initialized, and the filter initial state is set to some fixed value. In this case, the following initial covariance

and state values are assigned:

$$P_{ss}(t_i) = P_{si} \tag{J.1}$$

$$P_{fs}(t_i) = 0 \tag{J.2}$$

$$P_{ff}(t_i) = 0 \tag{J.3}$$

$$x_s(t_i) = P_{si}^{\frac{1}{2}} u_s \tag{J.4}$$

$$\hat{x}_f(t_i) = \hat{x}_{fi} \tag{J.5}$$

where t_i is the initial time of each run and u_s is a zero-mean, unit-variance random vector (3).
Thus, a randomized initial state condition is used, according to statistics provided by the user.

Mode 2 assumes that the simulation starts at some point well after the filter has started
operating, so that the filter state estimate is a fully developed random process. In this mode the
filter "initial state" is set equal to the corresponding system state $A_{fs} x_s$ plus an additional random
error, e_f. The following equations describe Mode 2:

$$P_{ss}(t_i) = P_{si} \tag{J.6}$$

$$P_{fs}(t_i) = A_{fs} P_{si} \tag{J.7}$$

$$P_{ff}(t_i) = A_{fs} P_{si} A_{fs}^T + P_{ei} \tag{J.8}$$

$$x_s(t_i) = P_{si}^{\frac{1}{2}} u_s \tag{J.9}$$

J-3

$$\hat{\mathbf{x}}_f(t_i) = \mathbf{A}_{fs}\mathbf{x}_s(t_i) + \mathbf{P}_{ei}^{\frac{1}{2}}\mathbf{u}_f \qquad (\text{J.10})$$

where $\mathbf{A}_{fs}$ represents the mapping matrix for correlation between the filter and truth models. For the GPS/INS problem, MSOFE used Mode 2. The same random initialization process was used by MATSOFE. MATSOFE is capable of operating under either mode just as MSOFE. At this time there is no switch in initcond.m to establish a specific initial conditions mode; the designer must determine which mode is appropriate for his needs and utilize it.

J.5 Running MATSOFE

The MATSOFE extended Kalman filtering algorithm begins with the command matekf2. Once this command is entered no other action is required of the user. The time consuming portion of the simulation is initially setting up the models. The algorithm progresses from the top-level script file, matekf2.m and first establishes the simulation controls in setup2.m. This is the first and one of the most important actions during the entire simulation. The subroutine, setup2.m, contains the initial and final run times, the number of runs, the number of measurements and the respective measurement rates, output rates, constants, noise power spectral density values, and initial values.

The algorithm then steps to the subroutine initrun2.m. This m-file sets the sizes for all MAT-SOFE vector arguments; establishes the initial conditions for Pf, Pei, Ps, xf, and xs; computes the initial trajectory values; and provides the initial values of the dynamics matrices. The subroutine usrout.m is also called to record the initial conditions of all variables of interest.

The trajectory program, mcirc2.m, computes the kinematic quantities for a vehicle traveling along a minor circle path. The trajectory variables are provided for use in the gps orbit routine, the observation matrices, and the dynamics models. The subroutine cannot fly a meridian flight path. If a trajectory other than a minor circle is required, then the data file must be input from PROFGEN or some other flight profile generator. To date, only the trajectory generator mcircle from MSOFE has been included into the MATSOFE tool. The subroutine gps.m computes the earth-referenced position and velocity vectors of specified GPS satellites in the user's geographic frame coordinates. The drawback to this algorithm is that all GPS satellite orbits pass directly over the user's initial position. The satellite starting positions can be selected to provide a reasonable

GPS constellation over a period of a few hours. MATSOFE has included the code to calculate the different figures of merit for the Dilution of Precision: GDOP, PDOP, HDOP, and TDOP (17).

The algorithm then moves to simrun2.m which conducts a complete simulation. This script is the heart of the extended Kalman filter processing. MATSOFE uses the extended Kalman filter equations to provide an estimate of user position errors (15). The filter-computed covariance is propagated forward in time using the discrete version of the Riccati Equation (2.17).

J.6   M-File Description

The following discussion describes each m-file used in MATSOFE and its sequence in the program. This summary is focused upon the GPS/INS integration problem of Chapter IV in the MSOFE Users' Manual (3). Where applicable, each m-file is related to its MSOFE counterpart for additional clarity. The flow graph of the program is shown in Figure J.1. *The m-files which change for each specific problem are indicated by an asterisk.*

1. **matekf2.m**   The m-file, matekf2.m (matekf-MATLAB extended Kalman filter), is currently the top level of the MATSOFE program. When the models and initial conditions have been input for the user's specific problem, the command *matekf2* is entered to begin program execution. The m-file proceeds as follows: (1) clear all variables, (2) call setup2.m, (3) set irun=1, (4) call initrun2.m, (5) call simrun2.m, (6) call endrun. Steps 4-6 are followed until the desired number of runs are completed. At the termination of the program, the data is processed for plotting using endsim.m.

2. **setup2.m**∗   The m-file, setup2.m, allows the user to declare all the important simulation controls for the problem in a single location. Constants, the initial and final simulation times, number of runs, number of measurements, measurement update rates, output rates, and measurement matrix update rates are established in setup2.m. This m-file is only called once at the beginning of the first Monte Carlo run; only variables that do not change between runs should be established in this m-file. The m-file, setup2.m, is analogous to MSOFE_IN Group 1 and usofe.for subroutine UBLOCK.

3. **initrun2.m**   The m-file, initrun2.m, initializes the parameters for each run. These variables are reset before each run. In initrun2.m, the time is set to the initial time and the first

measurement update, synchronous output, and propagation update is set. The m-file then calls out sizes.m, initcond.m, trajectory.m, newprop.m, and usrout.m.

4. **sizes.m∗**   This m-file establishes the model sizes for the entire simulation. This m-file is similar to the USOFE file SIZES.

5. **initcond.m∗**   The m-file, initcond.m, sets the initial conditions for a single Monte Carlo run. This m-file is analogous to MSOFE_IN Groups IV and V.

6. **trajectory.m∗**   The m-file, trajectory.m, is the top level m-file for calculating satellite and vehicle trajectories. The m-file calls out mcirc2.m and gps.m. This m-file is similar to TRAJSYS in USOFE.

7. **mcirc2.m∗**   The m-file, mcirc2.m, computes the vehicle position at each increment in time. This m-file is similar to MCIRCLE in usofe.for.
   Note: This routine is only for implementation of the GPS/INS Integration Problem of Chapter 4 of (3).

8. **gps.m∗**   The m-file, gps.m, calculates the earth-referenced position and velocity vectors of specified gps satellites in the user's geographic frame coordinates. This algorithm assumes earth-fixed circular orbits passing directly over the user's initial position. The m-file, newmeas.m, calculates all of the Dilutions of Precision, so that the system designer can set up the GPS initial conditions to employ whatever GDOP is necessary for his problem. This m-file is similar to GPSSAT in usofe.for.

9. **newprop.m∗**   The m-file, newprop.m, establishes the Fs, Ff, qs, and qf matrices [see Equation (2.11)]. The EKF theory is given in Chapter 2.The m-file, newprop.m, is called once during initrun.m to set the variables, and then called again before each propagation to reset the time-varying components. The MSOFE files ERRDYN, FCQSYS, and FCQFIL are similar to this m-file.

10. **usrout.m∗**   The m-file, usrout.m, is the primary subroutine for data storage throughout a single simulation run. This program is called from various points in the algorithm and allows the user to save variables stored in this subroutine to be written out to a file. This MATSOFE subroutine is designed to overwrite itself to save space; therefore, there is no generic CTOM or DTOM as in MSOFE for generating output files.

11. **simrun2.m**   The m-file, simrun2.m, is the control for a single run of the extended Kalman filter program.

12. **propagate.m**   The m-file, propagate.m, controls the propagation cycle for the extended Kalman filter. This m-file is called anytime there is a need to move forward in time. The m-file calls out trajectory.m, newprop.m, fsprop2.m, ffprop2.m, and Pfprop.m.

13. **fsprop2.m**   The m-file, fsprop2.m, propagates the system truth model vector, xs, to the next sample time [see Equation (2.16)].

14. fsform2.m*   The m-file, fsform2.m, establishes the equation to be propagated forward in time.

15. **qdform.m**   The m-file, qdform.m, takes the current value of qs and the Phis matrices and forms the equivalent discrete time measurement noise.

16. **ffprop2.m**   The m-file, ffprop2.m, propagates the xf vector to the next sample time [see Equation (2.16)].

17. **ffform2.m***   The m-file, ffform2.m, establishes the equation to be propagated forward in time.

18. **Pfprop.m**   The m-file, Pfprop.m, propagates the filter covariance, Pf from time to tnext [see Equation (2.17)]. Unlike the previous propagations (xf and xs) of the state covariances, this subroutine propagates the covariance using a discrete-time evaluation rather than the integration routines used in ffprop2.m and fsprop2.m.

19. **meas.m**   The m-file, meas.m, determines the complete measurement update cycle. This m-file calls out nummeas.m, premeas.m, calmeas.m, and postmeas.m.

20. **nummeas.m**   The m-file, nummeas.m, calculates how many measurements take place at the current measurement time depending on the relative sampling rates of the measurements.

21. **premeas.m**   The m-file, premeas.m, calls out newmeas.m and writes the necessary values of the measurement matrix to usrout.m.

22. **newmeas.m***   The m-file, newmeas.m, forms the new matrices needed to create the new zs truth model measurement, and the new xf and Pf. In order to form the necessary measurement matrices Hs, hs, Hf, hf, Rs, and Rf [see Equations (2.11), (2.13), (2.14), (3.19) and (3.20)] are computed in this m-file.

23. **calmeas.m**   The m-file, calmeas.m, calculates the measurement updates for the measurements that require updating as determined by nummeas.m [see Equations (2.21), (2.22), (2.23)]. The index values for the measurements to be updated are stored in wmeas, a vector of meas.m indices.

24. **postmeas.m**   The m-file, postmeas.m, holds all the events that occur at the time moment directly after a measurement update has occurred; all events that take place specifically at time ti-plus are called from this subroutine.

25. **endrun.m**   The m-file, endrun.m, is an output file called at at the end of each run, which allows the user to consolidate the data from each run.

26. **endsim.m∗**   The m-file, endsim.m, is used to store the computed statistical values of the variables of interests.

27. **output.m∗**   The m-file, output.m, is used to construct the desired plots of the stored data.

When real data is used as input files, several changes are required. Since the truth models are replaced by actual system outputs, the Fs and Qs matrices are no longer needed. These models are deleted from the m-files in newprop.m. Also the fsprop.m and fsform.m m-files are no longer necessary to propagate the system truth model.

Other m-files no longer needed are endrun.m and endsim.m. With real data there is only one run per data set and no processing of output variables for statisitics is required; only the residuals, position, and velocity outputs are collected in usrout.m for plotting. Another m-file no longer necessary is the mcirc2.m file, which provides the flight trajectory information. All of this data is contained in the real data files.

Several m-files are added to the routine to allow the use of real data. First of all, two INS routines are added for input of the LN-94 information. These two files are insdata.m and ltest. The file, ltest, is strictly a data file collected off of the 1553 data bus and it is read into the desired MATLAB format with the m-file insdata.m. The m-file, insdata.m, converts the binary format of the LN-94 to a MATLAB readable file.

The GPS data is processed for use in the EKF with several files. First, the data is collected into a data file from the XR-4PC or XR-5PC. The data is then converted into files for use in MATSOFE

with the master m-file navls.m. The data is read into large vectors using the m-file, convert.m, which is called out from navls.m. The function, process.m, takes the raw data and calculates satellite position in ECEF coordinates and applies the necessary corrections to the pseudo-ranges. Several actions then take place in navls.m. A position is calculated using the batch least squares method. The GDOP is computed and the number of satellites is checked to determine if a bad measurement has been received. Bad measurements are identified by a consistency check which determines the error between the actual range and pseudorange. If a bad measurement is identified, that satellite is not used in the navigation solution computations. The ECEF coordinates are transformed to Litton ECEF coordinates and the values converted from metric to English units. When changing models for the various GPS/INS runs, the files newprop.m, newmeas.m, initcond.m, setup2.m, and sizes.m were the only ones which required change.

Bibliography

1. Britting, Kenneth R. *Inertial Navigation Systems Analysis.* New York: Wiley-Interscience, 1971.

2. Canadian GPS Associates. *Guide to GPS Positioning.* ISBN: 0-920-114-73-3, Fredericton, New Brunswick: University of New Brunswick Graphic Services, May 1987.

3. Carlson, Neal and Stanton Musick. *User's Manual for a Multimode Simulation for Optimal Filter Evaluation (MSOFE).* AFWAL-TR-88-1138, Wright-Patterson AFB OH: Avionics Lab, AFWAL/AARN-2, April 1990.

4. Cox, D. B. "Integration of GPS with Inertial Navigation Systems," *Global Positioning System.* Papers Published in *Navigation*, Volume I, The Institute of Navigation, Washington D.C., 30 Sep 1987.

5. Department of the Air Force. *Specification for USAF F-15 Inertial Navigation Set.* FNU 85-1, Revision A. Aeronautical Systems Division, Wright Patterson AFB, OH, 1 May 1982.

6. Evans, Curtis D. and Brian J. Bohenek. *GPS Analysis* EENG-644 Class Project. Air Force Institute of Technology, Wright Patterson AFB OH, 3 June 1994.

7. Griffin, Gordon and others. *The Creation and Validation of a Matlab Tool As A Possible Alternative to MSOFE.* EENG-735 Class Project. Air Force Institute of Technology, Wright Patterson AFB OH, 3 June 1994.

8. Knudsen, L. *Performance Accuracy (Truth Model/Error Budget) Analysis for the LN-93 Inertial Navigation Unit.* Technical Report, Litton Guidance and Control Systems, January 1985. DID No. DI-S-21433 B/T:CDRL No. 1002.

9. Lewantowicz, Zdzislaw H. and Danny W. Keen. "Graceful Degradation of GPS/INS Performance With Fewer Than Four Satellites," *The Institute of Navigation, National Technical Meeting* (January 1991).

10. Hansen, Neil P. *Incorporation of Carrier Phase Global Positioning System Measurements into the Navigation Reference System for Improved Performance.* MS Degree Thesis, AFIT/GE/ENG/93D-40. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993 (AD-A274136).

11. Hirning, James L. *Optimal Kalman Filter Integration of a Global Positioning System Receiver and an LN-94 Inertial Navigation System.* MS Degree Thesis, AFIT/GE/ENG/90S-02. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993 (AD-A227222).

12. Knight, Donald T. "Achieving Modularity With Tightly-Coupled GPS/INS". *IEEE Position, Location, and Navigation Symposium.* 426-432. New York, IEEE Press, 1992.

13. *Matlab.* Version 4.1. Computer software. The MathWorks, Inc. 24 Prime Park Way, Natick, MA 01760. June 1993.

14. Maybeck, Peter S. *Stochastic Models, Estimation, and Control.* Volume 1. New York: Academic Press, Inc., 1979.

15. Maybeck, Peter S. *Stochastic Models, Estimation, and Control.* Volume 2. New York: Academic Press, Inc., 1982.

16. Maybeck, Peter S. *Stochastic Models, Estimation, and Control.* Volume 3. New York: Academic Press, Inc., 1982.

17. Milliken, R.J. and C.J. Zoller. "Principle of Operation of NAVSTAR and System Characteristics," *Global Positioning System, Papers Published in Navigation*, Volume I. Alexandria, VA 22314, 1980.

18. Mosle, William B. *Detection, Isolation, and Recovery of Failures in an Integrated Navigation System*. MS Degree Thesis, AFIT/GE/ENG/93D-28. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993 (AD-A274056).

19. Musick, Stanton H. *PROFGEN - A computer Program for Generating Flight Profiles*. Technical Report, Air Force Avionics Laboratory, WPAFB, Ohio, November 1976. AFAL-TR-76-247, DTIC ADA034993.

20. Negast, William Joseph. *Incorporation of Differential Global Positioning System Measurements Using an Extended Kalman Filter for Improved Reference System Performance*. MS Degree Thesis, AFIT/GE/ENG/91D-41. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991 (AD-A243742).

21. Riggins, Robert. *EENG 635 Inertial Navigation Subsystems Course Notes*. Air Force Institute of Technology, Wright Patterson AFB OH, January 1994.

22. Stacey, Richard D. *A Navigation Reference System (NRS) Using Global Positioning System (GPS) and Transponder Aiding*. MS Degree Thesis, AFIT/GE/ENG/91M-04. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1991 (AD-A238890).

23. Strang, Gilbert. *Linear Algebra and Its Applications*. Third Edition. San Diego: Harcourt Brace Jovanovich, Publishers, 1988.

24. Tazartes, D.A. and J.G. Mark "Integration of GPS Receivers Into Existing Inertial Navigation Systems," *Navigation: Journal of The Institute of Navigation* Vol. 34, No. 2, 815 Fifteenth Street, Suite 832, Washington, D.C. 20005, Summer 1988.

25. *XR4-PC GPS Receiver Installation and Operating Manual*. Navstar Ltd. Daventry, Northants NN11 5PJ, England, 2 September 1990.

26. *XR5-PC GPS Receiver Installation and Operating Manual*. Navstar Ltd. Daventry, Northants NN11 5PJ, England, 20 September 1993.

27. Vasquez, Juan R. *Detection of Spoofing, Jamming, or Failure of a Global Positioning System (GPS)*. MS Thesis, Air Force Institute of Technology, WPAFB, OH, December 1992.

28. Van Dierendonck, A.J. and others. "The GPS Navigation Message," *Navigation: Journal of The Institute of Navigation* Vol. 34, No. 2, 815 Fifteenth Street, Suite 832, Washington, D.C. 20005, Summer 1988.

Vita

Captain Curtis D. Evans hails from Nettie, West Virginia. In December 1983, Captain Evans enlisted in the United States Air Force. After serving two years as a technician and operator on electronic countermeasures and bomb scoring radar, Captain Evans was accepted into the Airmen Education and Commissioning Program (AECP). He attended the University of Missouri-Rolla and obtained his Bachelors of Science Degree in Electrical Engineering. After completing Officer Training School, Captain Evans was assigned to the Proven Aircraft Division at Kelly Air Force Base, Texas. In 1992, Captain Evans joined the F-5 Technical Coordination Group and also received his Masters Degree in Business Administration from Webster University. From June 1993 to December 1994, Captain Evans attended the Air Force Institute of Technology where he was awarded the degree of Masters of Science in Electrical Engineering. Captain Evans is married to Shonda (Fowler) Evans of Tioga, West Virginia and they have one son, Austin. Captain Evans is currently assigned to the Avionics Laboratory, Wright-Patterson AFB, OH.

Permanent address: Box 37AA Nettie, WV

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1994	Master's Thesis

4. TITLE AND SUBTITLE

THE DESIGN AND ANALYSIS OF INTEGRATED
NAVIGATION SYSTEMS USING REAL
INS AND GPS DATA

5. FUNDING NUMBERS

6. AUTHOR(S)

Curtis D. Evans

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/94D-08

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

WL/AAAS
Avionics Directorate
Wright-Patterson AFB, OH 45433-6583

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

An extended Kalman filter (EKF) is used to combine the information obtained from a Global Positioning System (GPS) receiver and an Inertial Navigation System (INS) to provide a navigation solution. This research compares the results of an actual tightly-coupled GPS/INS integrated system with those obtained in computer simulation. Differences between the simulation and actual hardware results, such as navigation solutions and tuning values, are shown in the research. An analysis of various GPS models was studied to determine the best representation for a system truth model. Measurement updates and their effects upon position accuracies was considered and preliminary research was conducted to determine when a feedback implementation was advantageous over a feedforward approach. The results suggest that the computer simulations being used at the Air Force Institute of Technology (AFIT) produce better navigation solutions than are actually possible with real hardware. Alternate system truth models were found which would estimate the errors of an actual system more accurately with less computational loading.

14. SUBJECT TERMS

Inertial Navigation System, INS, GLobal Positioning System, GPS,
Kalman Filter, MSOFE

15. NUMBER OF PAGES

242

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL